

CONFERENCES IN RESEARCH AND PRACTICE IN
INFORMATION TECHNOLOGY

VOLUME 101

DATA MINING AND ANALYTICS 2009
(AusDM'09)



AUSTRALIAN
COMPUTER
SOCIETY



DATA MINING AND ANALYTICS 2009 (AusDM'09)

Proceedings of the
Eighth Australasian Data Mining Conference (AusDM'09),
Melbourne, Australia, 1-4 December 2009

Paul J. Kennedy, Kok-Leong Ong, and Peter Christen,
Eds.

Volume 101 in the Conferences in Research and Practice in Information Technology Series.
Published by the Australian Computer Society Inc.



Published in association with the ACM Digital Library.

Data Mining and Analytics 2009 (AusDM'09). Proceedings of the Australasian Data Mining Conference 2009, Melbourne, Australia, 1-4 December 2009

Conferences in Research and Practice in Information Technology, Volume 101.

Copyright © 2009, Australian Computer Society. Reproduction for academic, not-for-profit purposes permitted provided the copyright text at the foot of the first page of each paper is included.

Editors:

Paul J. Kennedy
Faculty of Engineering and Information Technology
University of Technology, Sydney
Broadway, NSW, 2007, Australia
E-mail: paulk@it.uts.edu.au

Kok-Leong Ong
School of Information Technology
Deakin University
Burwood, Victoria 3125, Australia
E-mail: leong@deakin.edu.au

Peter Christen
School of Computer Science
ANU College of Engineering and Computer Science
The Australian National University
Canberra ACT 0200 Australia
E-mail: peter.christen@anu.edu.au

Series Editors:

Vladimir Estivill-Castro, Griffith University, Queensland
Simeon Simoff, University of Western Sydney, NSW
crpit@infoeng.flinders.edu.au

Publisher: Australian Computer Society Inc.
PO Box Q534, QVB Post Office
Sydney 1230
New South Wales
Australia.

Conferences in Research and Practice in Information Technology, Volume 101
ISSN 1445-1336
ISBN 978-1-920682-82-8

Printed November 2009 by Deakin Print Services, Deakin University, 221 Burwood Highway, Burwood, VIC 3125.

The *Conferences in Research and Practice in Information Technology* series aims to disseminate the results of peer-reviewed research in all areas of Information Technology. Further details can be found at <http://crpit.com/>.

Table of Contents

Proceedings of the Australasian Data Mining Conference 2009, Melbourne, Australia, 1-4 December 2009

Preface	vii
Programme Committee	ix
AusDM Sponsors	xi

Keynote Papers

What Data Mining Can Discover From Your Face ... No More Lying About Your Age!	3
<i>Kate Smith-Miles</i>	
Towards Web Search Engine Scale Data Mining	5
<i>Jian Pei</i>	
Credit Scoring and Data Mining	7
<i>Ross Gayler</i>	

Research Papers

Pattern Mining I

Efficiently Mining Frequent Subpaths	11
<i>Sumanta Guha</i>	
Distributed Association Rule Mining with Minimum Communication Overhead	17
<i>Md. Golam Kaosar, Zhuojia Xu, and Xun Yi</i>	

Clustering

Applying Clustering and Ensemble Clustering Approaches to Phishing Profiling	25
<i>John Yearwood, Dean Webb, Liping Ma, Peter Vamplew, Bahadorreza Ofoghi, and Andrei Kelarev</i>	
Clustering Interval-valued Data Using an Overlapped Interval Divergence	35
<i>Yongli Ren, Yu-Hsn Liu, Jia Rong, and Robert Dew</i>	
Reference Point Transformation for Visualization	43
<i>Cheng G. Weng and Josiah Poon</i>	

Pattern Mining II

FlowRecommender: A Workflow Recommendation Technique for Process Provenance	55
<i>Ji Zhang, Qing Liu, and Kai Xu</i>	
Negative-GSP: An Efficient Method for Mining Negative Sequential Patterns	63
<i>Zhigang Zheng, Yanchang Zhao, and Longbing Cao</i>	

Non-Redundant Rare Itemset Generation	69
<i>Yun Sing Koh and Russel Pears</i>	
<hr/>	
Applications I	
<hr/>	
Monetising User Generated Content Using Data Mining Techniques	75
<i>Yu-Hsn Liu, Yongli Ren, and Robert Dew</i>	
QUEST: Discovering Insights from Survey Responses	83
<i>Girish K. Palshikar, Shailesh S. Deshpande, and Savita S. Bhat</i>	
<hr/>	
Applications II	
<hr/>	
Discovering Inappropriate Billings with Local Density-based Outlier Detection Method	93
<i>Yin Shan, D. Wayne Murray, and Alison Sutinen</i>	
Predictive Analytics That Takes in Account Network Relations: A Case Study of Research Data of a Contemporary University	99
<i>Ekta Nankani and Simeon Simoff</i>	
Kernel-based Principal Components Analysis on Large Telecommunication Data	109
<i>Takeshi Sato, BingQuan Huang, Guillem Lefait, Tahar Kechadi, and Brian Buckley</i>	
<hr/>	
Time Series Data	
<hr/>	
SparseDTW: A Novel Approach to Speed up Dynamic Time Warping	117
<i>Ghazi Al-Naymat, Sanjay Chawla, and Javid Taheri</i>	
HDAX: Historical Symbolic Modelling of Delay Time Series in a Communications Network	129
<i>Hooman Homayounfard and Paul J. Kennedy</i>	
A Query Based Approach for Mining Evolving Graphs	139
<i>Andrey Kan, Jeffrey Chan, James Bailey, and Christopher Leckie</i>	
<hr/>	
Complex Data	
<hr/>	
Mining Minimal Constrained Flow Cycles from Complex Transaction Data	151
<i>Meng Xu and Michael Bain</i>	
Studying Genotype-Phenotype Attack on k-Anonymised Medical and Genomic Data	159
<i>Muzammil Mirza Baig, Jiuyong Li, Jixue Liu, and Hua Wang</i>	
<hr/>	
Graph Mining	
<hr/>	
Building a Generic Graph-based Descriptor Set for Use in Drug Discovery	167
<i>Phillip Lock, Nicolas Le Mercier, Jiuyong Li, and Markus Stumptner</i>	
Single Document Semantic Spaces	175
<i>Jorge Villalon and Rafael A. Calvo</i>	
Efficient Mining of Top-k Breaker Emerging Subgraph Patterns from Graph Datasets	183
<i>Min Gan and Honghua Dai</i>	
Edge Evaluation in Bayesian Network Structures	193
<i>Saaïd Baraty and Dan A. Simovici</i>	
Author Index	201

Preface

We are delighted to welcome you to the Eighth Australasian Data Mining Conference (AusDM'09) being held this year in Melbourne, Victoria in conjunction with 22nd Australian Joint Conference on Artificial Intelligence and the Fourth Australian Conference on Artificial Life (ACAL'09). AusDM started in 2002 and is now the annual flagship meeting for data mining and analytics professionals in Australia. Both scholars and practitioners present the state-of-the-art in the field. Endorsed by the peak professional body, the Institute of Analytics Professionals of Australia, AusDM has developed a unique profile in nurturing this joint community. The conference series has grown in size each year from early workshops held in Canberra (2002, 2003), Cairns (2004), Sydney (2005, 2006), the Gold Coast (2007) and Glenelg (2008). This year's event has been supported by

- Togaware, again hosting the website and the conference management system, coordinating the review process and other essential expertise;
- The University of Melbourne and Monash University for providing the venue, registration facilities and various other support;
- the Institute of Analytics Professionals of Australia (IAPA) for facilitating the contacts with the industry;
- the ARC Research Network on Data Mining and Knowledge Discovery, for providing financial support;
- the Australian Computer Society, for publishing the conference proceedings;
- Tiberius Data Mining for supporting the AusDM 2009 Analytic Challenge, providing the prize of \$1000 and hosting the competition website;
- data mining students from Deakin University for their local support.

The conference program committee reviewed 49 submissions, out of which 22 submissions were selected for publication and presentation. This was an acceptance rate of 44.8%. AusDM follows a rigid blind peer-review process and ranking-based paper selection process. All papers were extensively reviewed by at least three referees drawn from the program committee. We would like to thank all those who submitted their work to the conference. We will continue to extend the conference format to be able to accommodate more presentations.

In addition, three keynote speakers were invited. Professor Kate Smith-Miles from Monash University talked about What Data Mining Can Discover From Your Face ...No More Lying About Your Age!; Associate Professor Jian Pei of Simon Fraser University, Canada gave a talk about Towards Web Search Engine Scale Data Mining; and Dr Ross Gayler from Veda Advantage in Melbourne talked about Credit Scoring and Data Mining.

The conference also included the inaugural AusDM data mining competition: the AusDM Analytic Challenge “Ensembling” which looked at the problem of combining individual models to produce more accurate predictions. We would like to thank Tiberius Data Mining for supporting the competition and Netflix for providing the datasets as well as all the competition participants.

We were happy to host a special session on Visual Analytics, led by Assoc. Prof. Seok-Hee Hong, which aimed to foster links between this exciting emerging field and data mining.

We would like to extend our special thanks to the program committee members. The final quality of selected papers depends on their efforts. The review cycle runs on a very tight schedule and we would like to thank all reviewers for their commitment and professionalism.

Paul J. Kennedy, University of Technology, Sydney
Kok-Leong Ong, Deakin University
Peter Christen, The Australian National University
Organisers of AusDM 2009
December, 2009

Programme Committee

Programme Chairs

Paul J. Kennedy, University of Technology, Sydney
Kok-Leong Ong, Deakin University

Conference Chairs

Peter Christen, The Australian National University
Jiuyong Li, University of South Australia, Adelaide

Competition Chair

Phil Brierley, Tiberius Data Mining, Melbourne

Conference Steering Committee Chairs

Simeon Simoff, University of Western Sydney
Graham Williams, Australian Taxation Office, Canberra

Programme Committee

Longbing Cao (University of Technology, Sydney, Australia)
Xuan-Hong Dang (University of Melbourne, Victoria, Australia)
Vladimir Estivill-Castro (Griffith University, Queensland, Australia)
Ross Gayler (Veda Advantage, Melbourne, Australia)
Raj Gopalan (Curtin University of Technology, Perth, Australia)
Warwick Graco (Australian Taxation Office, Canberra, Australia)
Lifang Gu (Australian Taxation Office, Canberra, Australia)
Robert Hilderman (University of Regina, Canada)
Joshua Huang (University of Hong Kong, Hong Kong)
Warren Jin (CSIRO, Canberra, Australia)
Yun Sing Koh (Auckland University of Technology, New Zealand)
Gang Li (Deakin University, Victoria, Australia)
Bradley Malin (Vanderbilt University, Nashville, USA)
Arturas Mazeika (Free University of Bozen, Italy)
Richi Nayak (Queensland University of Technology, Brisbane, Australia)
Christine O’Keefe (CSIRO, Canberra, Australia)
Mehmet Orgun (Macquarie University, Sydney, Australia)
Tom Osborn (The Leading Edge, Sydney, Australia)
Robert Pearson (Canberra, Australia)
François Poulet (IRISA — Texmex, Rennes, France)
Richard Price (DSTO, South Australia, Australia)
Kate Smith-Miles (Monash University, Melbourne, Australia)
David Taniar (Monash University, Melbourne, Australia)
John Yearwood (University of Ballarat, Victoria, Australia)
Ting Yu (University of Sydney, Australia)
Huaifeng Zhang (Centrelink, Canberra, Australia)
Yanchang Zhao (University of Technology, Sydney, Australia)

Additional Reviewers

Aris Gkoulalas-Divanis (Vanderbilt University, Nashville, USA)
Bo Liu (University of Technology, Sydney, Australia)
Grigorios Loukides (Vanderbilt University, Nashville, USA)
Chao Luo (University of Technology, Sydney, Australia)

Tele Tan (Curtin University of Technology, Perth, Australia)
Yanshan Xiao (University of Technology, Sydney, Australia)
Yihao Zhang (Macquarie University, Sydney, Australia)
Zheng Zhigang (University of Technology, Sydney, Australia)
Ziye Zuo (University of Technology, Sydney, Australia)

AusDM Sponsors

We wish to thank the following sponsors for their contribution towards this conference.



<http://www.togaware.com>



<http://www.iapa.org.au>



ARC Research Network on Data Mining and Knowledge Discovery

<http://www.dmkd.flinders.edu.au>



<http://www.tiberius.biz/>

Conference Programme

Wednesday, 2 December, 2009

08:50 - 09:00 **AusDM 2009 Welcome**

09:00 - 09:50 **AI Keynote 1: Prof. Mark Bedau.**

09:50 - 10:40 **Session 1: Pattern Mining I**

09:50 - 10:15 EFFICIENTLY MINING FREQUENT SUBPATHS,
Sumanta Guha

10:15 - 10:40 DISTRIBUTED ASSOCIATION RULE MINING WITH MINIMUM
COMMUNICATION OVERHEAD,
Md. Golam Kaosar, Zhuojia Xu, Xun Yi

10:40 - 10:50 Tea break

10:50 - 12:05 **Session 2: Clustering**

10:50 - 11:15 APPLYING CLUSTERING AND ENSEMBLE CLUSTERING
APPROACHES TO PHISHING PROFILING,
John Yearwood, Dean Webb, Liping Ma, Peter Vamplew,
Bahadorreza Ofoghi, Andrei Kelarev

11:15 - 11:40 CLUSTERING INTERVAL-VALUED DATA USING AN
OVERLAPPED INTERVAL DIVERGENCE,
Yongli Ren, Yu-Hsn Liu, Jia Rong, Robert Dew

11:40 - 12:05 REFERENCE POINT TRANSFORMATION FOR VISUALIZATION,
Cheng Weng, Josiah Poon

12:05 - 12:55 **AusDM Keynote 1: Dr. Ross Gayler.**

12:55 - 14:10 Break

14:10 - 15:00 **AI Keynote 2: Prof. Ian Witten.**

15:00 - 16:15 **Session 3: Pattern Mining II**

15:00 - 15:25 FLOWRECOMMENDER: A WORKFLOW RECOMMENDATION
TECHNIQUE FOR PROCESS PROVENANCE,
Ji Zhang, Qing Liu, Kai Xu

15:25 - 15:50 NEGATIVE-GSP: AN EFFICIENT METHOD FOR MINING
NEGATIVE SEQUENTIAL PATTERNS,
Zhigang Zheng, Yanchang Zhao, Longbing Cao

15:50 - 16:15 NON-REDUNDANT RARE ITEMSET GENERATION,
Yun Sing Koh, Russel Pears

16:15 - 16:25 Tea break

16:25 - 17:40 **Special Session: Visual Analytics**

Led by Associate Professor Seok-Hee Hong

Thursday, 3 December, 2009

09:00 - 09:50 **AusDM Keynote 2: Prof. Kate Smith-Miles.**

09:50 - 10:40 **Session 4: Applications I**

09:50 - 10:15 MONETISING USER GENERATED CONTENT USING DATA
MINING TECHNIQUES,
Yu-Hsn Liu, Yongli Ren, Robert Dew

10:15 - 10:40 QUEST: DISCOVERING INSIGHTS FROM SURVEY RESPONSES,
Girish Palshikar, Shailesh Deshpande, Savita Bhat

10:40 - 10:50 Tea break

10:50 - 12:05 Session 5: Applications II

- 10:50 - 11:15 DISCOVERING INAPPROPRIATE BILLINGS WITH LOCAL DENSITY-BASED OUTLIER DETECTION METHOD,
Yin Shan, D. Wayne Murray, Alison Sutinen
- 11:15 - 11:40 PREDICTIVE ANALYTICS THAT TAKES IN ACCOUNT NETWORK RELATIONS: A CASE STUDY OF RESEARCH DATA OF A CONTEMPORARY UNIVERSITY,
Ekta Nankani, Simeon Simoff
- 11:40 - 12:05 KERNEL-BASED PRINCIPAL COMPONENTS ANALYSIS ON LARGE TELECOMMUNICATION DATA,
Takeshi Sato, BingQuan Huang, Guillem Lefait, Tahar Kechadi, Brian Buckley

12:05 - 12:55 AusDM Keynote 3: Assoc. Prof. Jian Pei.

12:55 - 14:10 Break

14:10 - 15:00 AI Keynote 3: Prof. Eamonn Keogh.

15:00 - 16:15 Session 6: Time Series

- 15:00 - 15:25 SPARSEDTW: A NOVEL APPROACH TO SPEED UP DYNAMIC TIME WARPING,
Ghazi Al-Naymat, Sanjay Chawla, Javid Taheri
- 15:25 - 15:50 HDAX: HISTORICAL SYMBOLIC MODELLING OF DELAY TIME SERIES IN A COMMUNICATIONS NETWORK,
Hooman Homayounfard, Paul Kennedy
- 15:50 - 16:15 A QUERY BASED APPROACH FOR MINING EVOLVING GRAPHS,
Andrey Kan, Jeffrey Chan, James Bailey, Christopher Leckie

16:15 - 16:25 Tea break

16:25 - 17:40 AusDM Analytics Challenge Presentation

Friday, 4 December, 2009

09:00 - 09:50 AI Keynote 4: Prof. Andries P. Engelbrecht.

09:50 - 10:40 Session 7: Complex Data

- 09:50 - 10:15 MINING MINIMAL CONSTRAINED FLOW CYCLES FROM COMPLEX TRANSACTION DATA,
Meng Xu, Michael Bain
- 10:15 - 10:40 STUDYING GENOTYPE-PHENOTYPE ATTACK ON K-ANONYMISED MEDICAL AND GENOMIC DATA,
Muzammil Mirza Baig, Jiuyong Li, Jixue Liu, Hua Wang

10:40 - 10:50 Tea break

10:50 - 12:30 Session 8: Graph Mining

- 10:50 - 11:15 BUILDING A GENERIC GRAPH-BASED DESCRIPTOR SET FOR USE IN DRUG DISCOVERY,
Phillip Lock, Nicolas Le Mercier, Jiuyong Li, Markus Stumptner
- 11:15 - 11:40 SINGLE DOCUMENT SEMANTIC SPACES,
Jorge Villalon, Rafael Calvo
- 11:40 - 12:05 EFFICIENT MINING OF TOP-K BREAKER EMERGING SUBGRAPH PATTERNS FROM GRAPH DATASETS,
Min Gan, Honghua Dai
- 12:05 - 12:30 EDGE EVALUATION IN BAYESIAN NETWORK STRUCTURES,
Saaïd Baraty, Dan Simovici

12:30 - 14:10 Break

14:10 - 15:00 Industry Presentation on Kaggle.com: Anthony Goldbloom.

15:00 - 15:25 AusDM Closing

KEYNOTE PAPERS

What Data Mining Can Discover From Your Face ...No More Lying About Your Age!

Kate Smith-Miles

School of Mathematical Sciences,
Faculty of Science, Monash University
Wellington Road, Clayton, Victoria 3800, AUSTRALIA
Email: kate.smith-miles@sci.monash.edu.au

Towards Web Search Engine Scale Data Mining

Jian Pei

School of Computing Science
Simon Fraser University
8888 University Drive, Burnaby, BC CANADA V5A 1S6
Email: jpei@cs.sfu.ca

Abstract

Data mining is one of the most critical driving technologies behind Web search engines. Web search engine scale data mining posts many grand challenges, ranging from efficiency and scalability to diversity and adaptability. In this talk, I will review our recent effort on mining a very large amount of data accumulated in one of the major commercial search engines. Particularly, we tackle the problem of context-aware search and query suggestion by employing statistical models. Moreover, we construct a very large statistical model (millions of states) from a very large amount of data (billions of sessions) by distributed data mining. I will also introduce some of our recent initiatives in Web mining.

Credit Scoring and Data Mining

Ross Gayler

Veda Advantage, Melbourne, AUSTRALIA

Abstract

Credit scoring is the use of predictive modelling techniques to support decision making in lending. It is a field of immense practical value that also supports a modest amount of academic research. Interestingly, the academic research tends not to be put into practice. This is not a result of insularity and arrogance on the part of the practitioners, but rather, of the practitioners having a better understanding of where they add value. This arises because credit scoring (and probably many other analytical applications) is dominated by shallow pragmatic issues rather than deep theoretical issues. In this talk I give examples of practical issues in credit scoring.

RESEARCH PAPERS

Efficiently Mining Frequent Subpaths

Sumanta Guha¹

¹ Computer Science & Information Management Program
Asian Institute of Technology
PO Box 4, Klong Luang, Pathumthani 12120, Thailand
Email: guha@ait.asia

Abstract

The problem considered is that of finding frequent subpaths of a database of paths in a fixed undirected graph. This problem arises in applications such as predicting congestion in network traffic. An algorithm based on Apriori, called AFS, is developed, but with significantly improved efficiency through exploiting the underlying graph structure, which makes AFS feasible for practical input path sizes. It is also proved that a natural generalization of the frequent subpaths problem is not amenable to any solution quicker than Apriori.

Keywords: AFS, Apriori, data mining, frequent subpath, frequent substructure, graph mining.

1 Introduction

Within the general problem of mining frequent patterns from a database of transactions, an area of some recent interest is where the transactions occur in a structured or semi-structured set. The structure considered often is that of a graph because objects under scrutiny in various applications can, in fact, be modeled as graphs, e.g., chemical compounds, web links, virtual communities, XML specifications and networks of different kinds. Finding frequent subgraphs of a database of graph transactions has been an area of particular activity. Apriori-based algorithms for this problem have been given, amongst others, by (Vanetik et al., 2002), (Inokuchi et al., 2000) and (Kuramochi and Karypis, 2001), while (Yan and Han, 2002) give an algorithm which uses a novel encoding scheme for graphs. See (Cook and Holder, 2006) for a survey of graph mining techniques.

The problem which we consider is a particular case of the problem of finding frequent subgraphs. In particular, in our case all transactions are paths in a fixed undirected graph, and we are interested in determining those paths in that graph which occur frequently as subpaths of the transaction paths. This is a natural problem to consider. For example, if each path in the database represents the route taken by an object such as a message or vehicle, then the frequent subpaths represent congested sections, or hot spots. Related work includes (Chen et al., 1998) and (Gudes and Pertsev, 2005), which both compute the *whole* paths themselves that are frequently traversed, rather than the frequently traversed shared parts which we consider (e.g., a set of paths may individually not be frequently traveled, but particular shared edges could well be congested).

Our algorithm is derived from Apriori (Agrawal and Srikant, 1994) as well. However, a simple-minded appli-

cation of Apriori – say, by treating paths as itemsets of vertices – fails because the feasibility of Apriori depends on transactions being of small size. However, paths in graphs arising from practical applications are not necessarily short (e.g., consider vehicular traffic in a city), and a straight Apriori-type solution runs into exponential complexity. Instead, we exploit the graph structure for a significant gain in efficiency which leads to a generally applicable solution, which we call AFS (Apriori for Frequent Subpaths). In fact, we analyze and compare the complexities of Apriori and AFS to prove a theoretical gain in efficiency from exponential in input size to low polynomial.

Next, we show that, interestingly, there is no possibility of similarly leveraging the graph structure to improve Apriori for a solution to a natural generalization of the frequent subpaths problem – that of finding so-called frequent strings of subpaths – because the general problem is equivalent in complexity to that of finding frequent itemsets.

2 Problem and Algorithm

2.1 Problem Statement

Let $G = (V, E)$ be an undirected graph with vertex set V and edge set E .

Here are some definitions related to paths in graphs which we'll use. A *path* P in G of length k from a vertex u to u' is a sequence (v_0, v_1, \dots, v_k) of vertices such that $v_0 = u$ and $v_k = u'$ and $(v_{i-1}, v_i) \in E$ for $i = 1, 2, \dots, k$. (We'll also allow the empty sequence $()$ to denote the empty path of undefined length.) A path Q in G is said to be a *subpath* of P , denoted $Q \triangleleft P$, if $Q = (w_0, w_1, \dots, w_{k'})$, where $(w_0, w_1, \dots, w_{k'})$ is a contiguous subsequence of (v_0, v_1, \dots, v_k) , i.e., if, for some i such that $0 \leq i \leq i + k' \leq k$, we have $w_0 = v_i, w_1 = v_{i+1}, \dots, w_{k'} = v_{i+k'}$. In this case, if $i = 0$, then Q is called a *prefix* subpath of P , and, if $i + k' = k$, then Q is called a *suffix* subpath of P . For a non-empty path $P = (v_0, v_1, \dots, v_k)$, *front*(P) denotes the first vertex v_0 and *tail*(P) denotes the suffix subpath (v_1, \dots, v_k) . A path (or, subpath) of length k will often be called a k -path (or, k -subpath).

Following are a few more definitions pertinent particularly to our problem. Let \mathcal{P} be a given set of paths in G . A path Q in G is said to have *support* $\text{support}(Q) = |\{P \in \mathcal{P} : Q \triangleleft P\}|$, i.e., the number of paths in \mathcal{P} of which Q is a subpath. Moreover, suppose a *minimum support* value min_sup is specified. If $\text{support}(Q) \geq \text{min_sup}$, then Q is said to be a *frequent subpath*.

The statement of the problem is now straightforward: *Given a set \mathcal{P} of paths in an undirected graph G , determine all frequent subpaths.*

See Figure 1 for an example of three paths in a grid graph.

Remark: In database terminology, \mathcal{P} is a database of transactions, where each transaction P is a path in a fixed graph G .

Copyright ©2009, Australian Computer Society, Inc. This paper appeared at the Eighth Australasian Data Mining Conference (AusDM 2009), Melbourne, Australia. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 101, Paul J. Kennedy, Kok-Leong Ong and Peter Christen, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

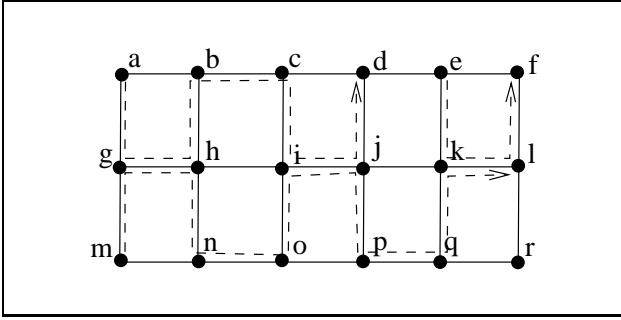


Figure 1: A grid graph with three paths indicated by directed broken lines. If $min_sup = 2$ then the frequent subpaths are (g) , (h) , (i) , (j) , (k) , (l) , (g, h) , (i, j) and (k, l) .

2.2 Apriori Algorithm

As our algorithm to find frequent subpaths is derived from the Apriori algorithm, and as we'll be comparing the complexities of the two, we'll first describe Apriori in some detail.

Let \mathcal{D} be a database of transactions, where each transaction $T \in \mathcal{D}$ is a subset of a set of all items \mathcal{I} . The support of an itemset $I \subset \mathcal{I}$ is $support(I) = |\{T \in \mathcal{D} : I \subset T\}|$. If $support(I) \geq min_sup$, for a specified value min_sup , then I is frequent. Following is pseudo-code for the Apriori algorithm to determine all frequent itemsets (adapted from (Agrawal and Srikant, 1994)).

Apriori

```

 $L_1 = \{\text{frequent 1-itemsets}\};$ 
for ( $k = 2; L_{k-1} \neq \emptyset; k++$ )
{
     $C_k = \text{join}(L_{k-1}, L_{k-1});$  // Generate candidates.
     $C_k = \text{prune}(C_k);$  // Prune candidates.
     $L_k = \text{checkSupport}(C_k);$  // Eliminate candidate
    // if support too low.
}
return  $\cup_k L_k;$  // Returns all frequent itemsets.

```

We discuss next the routines in the Apriori **for** loop and how all three are implemented using a function $subset(X, T)$, where X is a set of itemsets and T is an itemset, which returns the subset Y of X consisting of those itemsets which are contained in T (we'll discuss implementing $subset(X, T)$ itself later).

Firstly, $\text{join}(L_{k-1}, L_{k-1})$ generates all k -itemsets of the form $\{i_1, i_2, \dots, i_k\}$, where both $\{i_1, i_2, \dots, i_{k-1}\}$ and $\{i_1, i_2, \dots, i_{k-2}, i_k\}$ belong to L_{k-1} (note that itemsets are always assumed listed in lexicographic order), i.e., unions of pairs of itemsets in L_{k-1} both of whose members share the same first $k-2$ items. Secondly, $\text{prune}(C_k)$ deletes all $I \in C_k$ such that some $(k-1)$ -subset of I does not belong to L_{k-1} . It may be checked that *both* $\text{join}(L_{k-1}, L_{k-1})$ and $\text{prune}(C_k)$ are implemented by the following routine which uses $subset(L_{k-1}, *)$:

pruneJoin

```

 $C_k = \emptyset;$ 
for each itemset  $I = \{i_1, i_2, \dots, i_{k-1}\} \in L_{k-1}$ 
    for each item  $j \in \mathcal{I}$  such that  $j > i_{k-1}$ 
    {
         $I' = \{i_1, i_2, \dots, i_{k-1}, j\};$ 
        for each  $(k-1)$ -subset  $A$  of  $I'$ 
            if ( $subset(L_{k-1}, A) = \emptyset$ ) goto reject;
            // Reject  $I'$  if it has a  $(k-1)$ -subset
            // not belonging to  $L_{k-1}$ .
        add  $I'$  to  $C_k$ ;
    }
reject:
return  $C_k;$  // Returns  $\text{prune}(\text{join}(L_{k-1}, L_{k-1}))$ .

```

Finally, $\text{checkSupport}(C_k)$ counts the support of each itemset currently in C_k to eliminate those which are not frequent. It is straightforwardly implemented with the help of $subset(C_k, *)$:

checkSupport

```

 $L_k = \emptyset;$ 
for each  $I \in C_k$ 
     $I.count = 0;$ 
for each transaction  $T \in \mathcal{D}$ 
    {
         $C_T = \text{subset}(C_k, T);$ 
        for each  $I \in C_T$ 
             $I.count++;$ 
    }
for each  $I \in C_k$ 
    if ( $I.count \geq min\_sup$ ) add  $I$  to  $L_k$ ;
return  $L_k;$  // Returns members of  $C_k$  with support
    // at least  $min\_sup$ .

```

Therefore, when implementing Apriori the calls to join and prune in the **for** loop are replaced by a single call to pruneJoin , while checkSupport is implemented as above.

The function $subset(X, T)$ itself is implemented by first storing the itemsets of X in a trie (prefix tree) (Fredkin, 1960) \mathcal{T} on the "alphabet" \mathcal{I} of items ordered lexicographically, each itemset treated as an ordered string. (Agrawal and Srikant, 1994) actually use a particular implementation called a hash tree (Coffman Jr. and Eve, 1970), where pointers to children are stored in a hash table keyed on items at each internal node (the use of a hash tree in this case instead of a simple trie is justified by the typically large size of \mathcal{I}). See Figure 2 for an example.

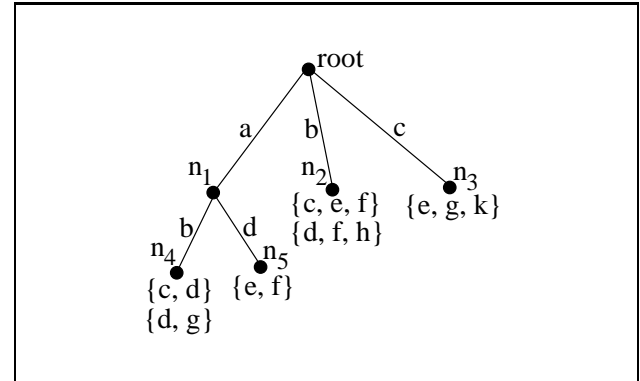


Figure 2: A hash tree \mathcal{T} storing a set of six 4-itemsets $X = \{\{a, b, c, d\}, \{a, b, d, g\}, \{a, d, e, f\}, \{b, c, e, f\}, \{b, d, f, h\}, \{c, e, g, k\}\}$, where each leaf can store at most two itemsets (only the suffix of an itemset following the prefix defined by the path to the leaf is stored).

The function $subset(X, T)$ is then executed by calling $\text{doSubset}(\text{root}(\mathcal{T}), T)$ using the recursive routine below:

doSubset($node, I$)

```

{
     $Y = \emptyset;$ 
    if (node is leaf) add  $\text{checkItemsets}(node, I)$  to  $Y$ ;
    // Function  $\text{checkItemsets}(node, I)$  returns those
    // itemsets stored at  $node$  that are contained in  $I$ .
    else if ( $I = \emptyset$ ); // Nothing is added to  $Y$ .
    else for each ( $i \in I$ )
        if ( $node.ch(i)$  exists)
            add  $i * \text{doSubset}(node.ch(i), \{j \in I : j > i\})$  to  $Y$ ;
            // For each item  $i \in I$  recurse on the corresponding
            // child of  $node$ . We denote by  $i * Z$  the union of  $i$ 
            // with each itemset in  $Z$ .
}

```

return $Y;$

}

For example, in Figure 2, $\text{doSubset}(\text{root}, \{a, b, c, d, e, f\})$ makes three recursive calls to doSubset with parameters $(n_1, \{b, c, d, e, f\})$, $(n_2, \{c, d, e, f\})$ and $(n_3, \{d, e, f\})$, respectively. The first of these in turn calls doSubset with parameters $(n_4, \{c, d, e, f\})$ and $(n_5, \{e, f\})$, while the second and third add $\{b, c, e, f\}$ and nothing, respectively, to the answer Y , etc.

Though various technical improvements in implementing Apriori have been suggested – see (Han and Kamber, 2005) for a discussion – we'll not consider them here, but use as our reference the basic implementation described above. This is in the interests of making an apples-to-apples comparison with AFS, whose basic implementation is described next.

2.3 Apriori for Frequent Subpaths

We present our algorithm AFS (Apriori for Frequent Subpaths) in a manner as similar as possible to that for Apriori in the previous section, so that it's easy to see exactly how the added structure in the setting of AFS helps make it more efficient.

AFS

```

 $L_0 = \{\text{frequent 0-subpaths}\};$ 
for ( $k = 1; L_{k-1} \neq \emptyset; k++$ )
{
   $C_k = \text{AFSextend}(L_{k-1});$  // Generate candidates.
   $C_k = \text{AFSprune}(C_k);$  // Prune candidates.
   $L_k = \text{AFScheckSupport}(C_k);$ 
  // Eliminate candidate if support too low.
}
return  $\cup_k L_k;$  // Returns all frequent subpaths.

```

The gain from the graph structure is first seen in generating candidates: we obtain C_k by simply extending each path in L_{k-1} by every edge incident on its last vertex (instead of potentially “joining” every pair of paths in L_{k-1}). This is justified as it may be seen that the set of k -paths obtained by so extending paths in L_{k-1} indeed contains L_k . Pruning is simpler as well because, after extending a path P in L_{k-1} to a k -path P' , the only $(k-1)$ -subpath of P' whose membership in L_{k-1} need be checked is its suffix $k-1$ -subpath. The reason is that P' has only two $k-1$ -subpaths: one prefix (P itself) and the other suffix.

E.g., in Figure 1, $(g, h) \in L_1$ would generate four extensions for inclusion in C_2 : (g, h, i) , (g, h, b) , (g, h, g) and (g, h, n) . Moreover, in the prune step, e.g., for (g, h, i) , only (h, i) has to be checked if it belongs to L_1 .

Both $\text{AFSextend}(L_{k-1})$ and $\text{AFSprune}(C_k)$ are implemented by the routine AFSpruneExtend below, which should be compared with the earlier pruneJoin routine for Apriori. AFSpruneExtend uses the function $\text{subpaths}(X, P)$, where X is a set of paths and T is a path, which returns the subset Y of X consisting of those paths which are subpaths of T . Function $\text{subpaths}(X, P)$, whose implementation we'll detail momentarily, is, of course, the counterpart of the earlier $\text{subset}(X, T)$.

AFSpruneExtend

```

 $C_k = \emptyset;$ 
for each path  $P = (v_0, v_1, \dots, v_{k-1}) \in L_{k-1}$ 
for each vertex  $v \in V$  adjacent to  $v_{k-1}$ 
{
   $P' = (v_0, v_1, \dots, v_{k-1}, v);$ 
  if ( $\text{subpaths}(L_{k-1}, (v_1, \dots, v_{k-1}, v)) = \emptyset$ )
    goto reject;
  // Reject  $P'$  if its suffix  $(k-1)$ -subpath
  // does not belong to  $L_{k-1}$ .

  add  $P'$  to  $C_k;$ 
}

```

reject:

```

}
return  $C_k;$  // Returns  $\text{ASFprune}(\text{ASFextend}(L_{k-1}))$ .

```

The routine AFScheckSupport is a near copy of its Apriori counterpart checkSupport .

AFScheckSupport

```

 $L_k = \emptyset;$ 
for each  $Q \in C_k$ 
   $Q.\text{count} = 0;$ 
for each path  $P \in \mathcal{P}$ 
{
   $C_P = \text{subpaths}(C_k, P);$ 
  for each  $Q \in C_P$ 
     $Q.\text{count}++;$ 
}
for each  $Q \in C_k$ 
  if ( $Q.\text{count} \geq \text{min\_sup}$ ) add  $Q$  to  $L_k;$ 
return  $L_k;$  // Returns members of  $C_k$  with support
  // at least  $\text{min\_sup}$ .

```

Therefore, when implementing AFS the calls to AFSextend and AFSprune in the for loop are replaced by a single call to AFSpruneExtend , while AFScheckSupport is implemented as above.

It's in implementing $\text{subpaths}(X, P)$ that we leverage the graph setting of AFS to huge gain over $\text{subset}(X, T)$ (we'll see the actual calculations in the next section). Paths in X are stored in a hash tree \mathcal{T} as well, exactly as for $\text{subset}(X, T)$. It's straightforward to use this tree of paths to determine which are prefix subpaths of P . Therefore, noting that a path in X is a subpath of P if and only if it is a prefix subpath of some suffix subpath of P , $\text{subpaths}(X, P)$ is implemented by calling $\text{doSubpaths}(\text{root}(\mathcal{T}), (w_0, w_1, \dots, w_k))$, where $P = (w_0, w_1, \dots, w_k)$.

```

doSubpaths( $\text{node}, \{w_0, w_1, \dots, w_k\}$ )
{
   $Y = \emptyset;$ 
  for ( $i = 0; i \leq k; i++$ )
    add doPrefixSubpaths( $\text{node}, (w_i, w_{i+1}, \dots, w_k)$ )
    to  $Y;$ 
  // Iteratively calls doPrefixSubpaths( $\text{node}, Q$ ) // on
  each suffix of  $Q$  of  $P = (w_0, w_1, \dots, w_k)$ .

  return  $Y$ 
}

```

Compare the following with doSubset .

```

doPrefixSubpaths( $\text{node}, Q$ )
{
   $Y = \emptyset;$ 
  if ( $\text{node}$  is leaf) add checkPrefixPaths( $\text{node}, Q$ ) to  $Y;$ 
  // Function checkPrefixPaths( $\text{node}, Q$ ) returns those
  // paths stored at  $\text{node}$  that are prefix subpaths of  $P$ .

  else if ( $Q = ()$ ); // Nothing is added to  $Y$ .
  else
    if ( $\text{node.ch}(\text{first}(Q))$  exists)
      add  $\text{first}(Q) * \text{doPrefixSubpaths}(\text{node.ch}(\text{first}(Q)),$ 
       $\text{tail}(Q))$  to  $Y;$ 
    // Descend from  $\text{node}$  along the path labeled by
    // successive vertices of  $Q$ . We denote by  $v * Z$  the
    // concatenation of  $v$  with each path in  $Z$ .

  return  $Y;$ 
}

```

For example, suppose the hash tree in Figure 2 represents a set of paths instead of itemsets. Then,

the call $\text{doSubpaths}(\text{node}, (a, b, c, d, e, f))$ spawns six iterations of the call doPrefixSubpaths with parameters $(\text{node}, (a, b, c, d, e, f))$, $(\text{node}, (b, c, d, e, f))$, \dots , $(\text{node}, (f))$, respectively. Each of the doPrefixSubpaths calls descends recursively from the root down a single path of T . E.g., the one with parameters $(\text{node}, (a, b, c, d, e, f))$ descends to n_4 to finally call $\text{doPrefixSubpaths}(n_4, (c, d, e, f))$, which adds (a, b, c, d) to the answer Y .

2.4 Complexity: AFS vs. Apriori

Consider Apriori first. The recursion in $\text{doSubset}(\text{node}, I)$ yields a Fibonacci-type recurrence in running time of $t(k) = t(k-1) + t(k-2) + \dots + t(1)$, if $I = \{i_1, i_2, \dots, i_k\}$, implying a time bound function of order exponential in the size of I , which we indicate by $O(\exp(|I|))$ (We ignore the cost of calls to $\text{checkItemsets}(\text{node}, I)$.) The size of the hash tree rooted at node is an obvious upper time bound as well on $\text{doSubset}(\text{node}, I)$.

Therefore, similar bounds apply to $\text{subset}(X, T)$ as well. In particular, $\text{subset}(C_k, T)$ and $\text{subset}(L_k, T)$, used to implement Apriori, are bounded in running time by $O(\min(\exp(|T|), \text{size_ht}(C_k)))$ and $O(\min(\exp(|T|), \text{size_ht}(L_k)))$, respectively, where $\text{size_ht}(X)$ denotes the size of the hash tree storing X .

It follows that the total time cost incurred by calls to pruneJoin from Apriori is

$$O(|J| \sum_k |L_k| \min(\exp(k), \text{size_ht}(L_k)))$$

(the expectation that on the average there will be $O(|J|)$ items greater than the last one in an itemset justifies the $|J|$ factor) and by those to checkSupport is

$$O(\sum_k (|C_k| + \sum_{T \in \mathcal{D}} \min(\exp(|T|), \text{size_ht}(C_k))))$$

Next, consider AFS. The routine $\text{doPrefixSubpaths}(\text{node}, Q)$ is bounded by time linear in $|Q|$ as the recursion descends from node along a path labeled by successive vertices of Q . The height of the hash tree rooted at node is a bound as well. Consequently, $\text{doSubpaths}(\text{node}, P)$ takes time $O(\min(|P|, \text{height}) + \min(|P| - 1, \text{height}) + \dots + \min(1, \text{height})) = O(\min(|P|^2, |P| \text{height}))$.

Therefore, $\text{subpaths}(C_k, P)$ runs in time bounded by $O(\min(|P|^2, |P| \text{height_ht}(C_k)))$, and $\text{subpaths}(L_k, P)$ in time bounded by $O(\min(|P|^2, |P| \text{height_ht}(L_k)))$, where $\text{height_ht}(X)$ denotes the height of the hash tree storing X , which represents a gain in efficiency over the corresponding Apriori routine $\text{subset}(X, T)$ from exponential to quadratic.

We have, therefore, that the total time cost incurred by calls to AFSextendJoin from AFS is

$$O(\sum_k |L_k| \min(k^2, \text{height_ht}(L_k)))$$

(we assume that on the average each vertex has $O(1)$ neighbors) and those to AFScheckSupport is

$$O(\sum_k (|C_k| + \sum_{P \in \mathcal{P}} \min(|P|^2, \text{height_ht}(C_k))))$$

Clearly, Apriori is vulnerable to exponential time worst-case behavior. In fact, it's evident from the complexity expressions for pruneJoin and checkSupport that the feasibility of applying Apriori lies in assuming that (a) the size of individual transactions in the database is $O(1)$, and (b) the size of C_k decreases rapidly with k . Fortunately, both assumptions are justified in various practical scenarios, e.g., market basket analysis.

In case of AFS though (a) is not a reasonable assumption: transactions in the database, i.e., paths in a graph, may not be short, or $O(1)$ in length. In practical applications, e.g., vehicles traveling in a network of roads, paths taken may even be of size comparable to that of the graph itself. However, we see from the last two expressions above that, even then, AFS has a worst-case behavior quadratic in the total length of the input paths, making it practically applicable.

Experimental Verification: The theoretical advantage of AFS can be tested in practical situations by using existing test data, or by generating random paths in large graphs, and then finding frequent subpaths using both Apriori (ignoring the graph structure and treating paths as itemsets of vertices) and AFS. We are currently in the process of setting up such experiments.

2.5 A Generalization and its Hardness

The intersection of a set of paths in an undirected graph G is not necessarily a path, but a union of paths. We'll call such an intersection a *string* of subpaths, or, simply, string. Therefore, a natural generalization of the frequent subpaths problem considered in the previous section is as follows: *Given a set \mathcal{P} of paths in an undirected graph G , determine all frequent strings of subpaths.*

For example, in Figure 1, $(g, h) \cup (i, j)$ and (k, l) are the two maximal frequent strings. Observe that knowing all frequent strings evidently implies knowing all frequent subpaths. However, the converse is not true – e.g., it's not possible to deduce from the fact that (g, h) , (i, j) and (k, l) are frequent subpaths in Figure 1, that $(g, h) \cup (i, j)$ is a frequent string. Therefore, the problem of finding frequent strings is at least as hard as that of finding frequent subpaths.

Surely, an Apriori-type algorithm may be implemented to find all frequent strings, but, interestingly, no improvement in efficiency over Apriori (as in AFS) can be expected because, as we'll see momentarily, the problem of finding frequent itemsets is equivalent to that of finding frequent strings. Firstly, we'll reduce the first problem to the second in time linear in the size of the input.

Let \mathcal{D} be a database of transactions, each transaction T being a subset of the set of all items J . Let G be the complete graph on the set of vertices $V = J$. Represent each transaction $T \in \mathcal{D}$, where $T = \{i_1, i_2, \dots, i_k\}$, by the path $P_T = (i_1, i_2, \dots, i_k)$, the items in T being in lexicographic order. It may be seen that, given the set of paths $\mathcal{P} = \{P_T : T \in \mathcal{D}\}$, the set of frequent strings corresponds exactly to the set of frequent itemsets for the database \mathcal{D} , which completes the reduction claimed and proves that finding frequent strings is at least as hard as finding frequent itemsets.

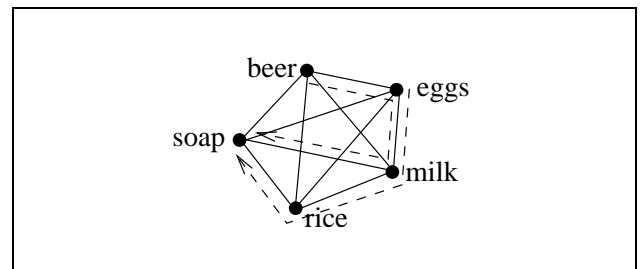


Figure 3: The database of two transactions $\{\text{beer}, \text{eggs}, \text{milk}, \text{soap}\}$ and $\{\text{eggs}, \text{milk}, \text{rice}, \text{soap}\}$ over the set of items $J = \{\text{beer}, \text{eggs}, \text{milk}, \text{rice}, \text{soap}\}$ is represented by two corresponding paths in the complete graph on J .

E.g., for the database of Figure 3, if $\text{min_sup} = 2$, then the one maximal frequent itemset is $\{\text{eggs}, \text{milk}, \text{soap}\}$ and the corresponding one maximal frequent string is $(\text{eggs}, \text{milk}) \cup (\text{soap})$.

We'll omit details here of the reduction in the opposite direction. The equivalence of the two problems means that there is no hope of leveraging the graph structure to find a more efficient variation of Apriori to determine frequent strings. However, this should not be an issue in practical applications where it is enough to simply identify the congested subpaths.

3 CONCLUSIONS

We have developed the AFS algorithm to find frequent subpaths which, though derived from Apriori, exploits the underlying graph structure for a gain in efficiency that makes it applicable to practical input sizes for this particular problem. We believe that similar improvements may be found for related problems, e.g., finding frequent subtrees of a collection of trees.

The development of a general framework in which to place the problem of finding frequent substructures of a collection of structures belonging to a family with certain given inheritance properties would be significant as well.

REFERENCES

- Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules. In *VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases*, pages 487–499.
- Chen, M. S., Park, J. S., and Yu, P. S. (1998). Efficient data mining for path traversal patterns. *IEEE Transactions on Knowledge and Data Engineering*, 10:209–221.
- Coffman Jr., E. G. and Eve, J. (1970). File structures using hashing functions. *Communications of the ACM*, 13:427–432.
- Cook, D. J. and Holder, L. B. (2006). *Mining Graph Data*. Wiley Inter-science.
- Fredkin, E. (1960). Trie memory. *Communications of the ACM*, 3:490–499.
- Gudes, E. and Pertsev, A. (2005). Mining module for adaptive xml path indexing. In *Proceedings of the 16th International Workshop on Database and Expert Systems Applications*, pages 1015–1019.
- Han, J. and Kamber, M. (2005). *Data Mining Concepts and Techniques, 2nd Ed.* Morgan Kaufmann.
- Inokuchi, A., Washio, T., and Motoda, H. (2000). An apriori-based algorithm for mining frequent substructures from graph data. In *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery (Lecture Notes In Computer Science, Vol. 1910)*, pages 12–23.
- Kuramochi, M. and Karypis, G. (2001). Frequent subgraph discovery. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, pages 313–320.
- Vanetik, N., Gudes, E., and Shimony, S. E. (2002). Computing frequent graph patterns from semistructured data. In *Proceedings of the 2002 IEEE International Conference on Data Mining*, pages 458–465.
- Yan, X. and Han, J. (2002). gspan: Graph-based substructure pattern mining. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, pages 721–724.

Distributed Association Rule Mining with Minimum Communication Overhead

Md. Golam Kaosar, Zhuojia Xu and Xun Yi

School of Engineering and Science, Victoria University, Australia

Victoria University PO Box 14428, Victoria 8001, Australia

md.kaosar@live.vu.edu.au, zhuojia.xu@live.vu.edu.au, xun.yi@vu.edu.au

Abstract

In distributed association rule mining algorithm, one of the major and challenging hindrances is to reduce the communication overhead. Data sites are required to exchange lot of information in the data mining process which may generates massive communication overhead. In this paper we propose an association rule mining algorithm which minimizes the communication overhead among the participating data sites. Instead of transmitting all itemsets and their counts, we propose to transmit a binary vector and count of only frequently large itemsets. Message Passing Interface (MPI) technique is exploited to avoid broadcasting among data sites. Performance study shows that the proposed algorithm performs better than two other well known algorithms known as Fast Distributed Algorithm for Mining Association Rules (FDM) and Count Distribution (CD) in terms of communication overhead.

Keywords: MPI, Data Mining, Association rule mining, AllGather, AllReduce.

1. Introduction

Though information technology (IT) is considered one of the greatest blessings of technology at current era, rapid inflation of information may explode the whole arena of IT if it is not supervised properly. Data mining is one of the means to utilize information by discovering underlying hidden useful knowledge from information. Among different approaches, association rule mining is one of the popular techniques for mining data. In this technique, an interrelation among different items in data is discovered by determining frequent large itemsets which are repeated more than a threshold number of times in the database. Association rule mining can enhance in extracting knowledge in various applications including advertisements, bioinformatics, database marketing, fraud detection, E-commerce, health care, security, sports,

telecommunication, web, weather forecasting, financial forecasting, etc. Data mining process can be characterized as centralized and distributed based on the location of data. In case of centralized data mining process, data is resided into a single site whereas in distributed process data is resided into multiple sites. The data may be owned by each site separately or an enormous amount of data may be distributed into multiple data sites.

In distributed ubiquitous computing environment lot of devices, sensors, terminals, equipments, computers, etc. are connected to each other through heterogeneous communication means in which minimization of bandwidth usage is considered as a major concern. To launch data mining applications in such an environment must require an algorithm which minimizes communication overhead.

Association rule mining process can be divided into two major tasks: (a) computation of all frequently large itemsets and (b) generation of all strong association rules which satisfy certain constraints. Since task (b) is considered as straightforward, most research efforts focus on task (a). In this paper we propose a distributed association rule mining algorithm to accomplish task (a) with the objective of minimizing communication overhead.

Significant amount of research work has been performed in association rule mining algorithms. Apriori algorithm proposed by R. Agrawal and R. Srikant (1994) is a classical and popular association rule mining algorithm which is suitable for centralized data mining. Due to the necessity of rapidly growing distributed computing environment, distributed mining algorithms become popular in the market. Distributed data mining algorithm is also necessary to ensure security and privacy in many other circumstances. R. Agrawal and J.C. Shafer (1996) propose a parallel and distributed association rule mining algorithm known as Count Distribution (CD). Main focus of CD is to reduce the communication overhead with the cost of redundant computation in all sites. This model is suitable for a system for which the computational capability dominates the communicational capability. Another algorithm, Fast Distribution Algorithm (FDM) of W. Cheung, J. Han, V. T. Ng, A. W. Fu, Y. Fu (1996), proposed to reduce the number of candidate sets generated in local sites, consequently reduces the communication overhead. It introduces local and

global pruning techniques to eliminate redundant computation. Both of these algorithms are discussed in section 3 of this paper in more detail.

Q. Ding, Q. Ding, W. Perrizo (2008) propose an efficient algorithm for mining association rules from spatial data for remote sensed imagery (RSI) data. In most association rule mining algorithms, binary relationship (presence or absence) in transactions are considered. But there are some efforts which consider the weight of the transactions too. A weighted association rule mining technique is introduced in K. Sun, F. Bai (2008). Weight of the association rule is measured by introducing a link-based model. A definition of weighted support is introduced and a weighted association rule mining (WARM) algorithm is proposed by F. Tao, F. Murtagh, M. Farid (2003). Both profit and purchased quantity are considered in mining transactional data in S. J. Yen, Y. S. Lee (2007). Different research efforts focus to accomplish different objectives but there is not much research work found which focus to minimize communication overhead in data distributed mining process. Therefore the proposed communication efficient algorithm might lead to bright possibility of deploying data mining applications in ubiquitous computing environment.

The CD algorithm is proposed to reduce the communication overhead without focusing much about the computational overhead. On the other hand DFM focuses on pruning in local and global level to reduce computation in the data sites. Therefore it is speculated that combination of these two algorithms might lead to a simple and efficient solution for association rule mining. Farther contemplation on the algorithms revealed that not all the count values of itemsets are necessary to be transmitted. We introduce an idea of determining the frequent itemsets first without exchanging the counts and then exchanging the counts of only those frequently large itemsets which are locally frequent in at least one site. Not only that, it is also possible to eliminate the redundant computation in each data site. Finally we come up with a new association rule mining algorithm idea, which might perform better than CD and FDM in terms of communication overhead.

Rest of the paper is organized as the following order: Section 2 describes relevant background information in brief. The proposed algorithm is described in section 3 while section 4 illustrates the performance analysis and comparison. Finally section 5 concludes the proposed algorithm.

2. Background

In this section some techniques and algorithms are discussed as background information in brief which are related to the proposed algorithm.

Message Passing Interface (MPI): This is a technique to exchange information among a number of

communicating nodes. It is especially suitable for mathematical functions like: summation or accumulation of a particular number which is to be calculated and distributed among nodes. This allows nodes to exchange the information without broadcasting; therefore, it reduces the communication overhead and communication round significantly. Detail of MPI can be found in R. Agrawal and J. Shafer (1996) and Argonne National Laboratory (MPI). Following example illustrates the functionality of MPI in brief.

Let us consider 8 nodes S_1, S_2, \dots, S_8 have their own count values to be summed and shared among themselves. One straightforward solution is to broadcast everyone's count to others in the common medium. This practice would require massive data transmission over the medium. Moreover, broadcasting is avoided due to many other reasons if alternatives are feasible. MPI, in this case, provides the best solution which is divided into two sub tasks: ReduceScatter and AllGather. In ReduceScatter, the sum of the counts is accumulated in a single predefined node whereas in AllGather, the sum is transmitted back to all nodes to be shared. Fig.1 shows how ReduceScatter accumulates a distributed value. It requires \log_2^N (3 in this case) steps to accumulate the result in the converging site.

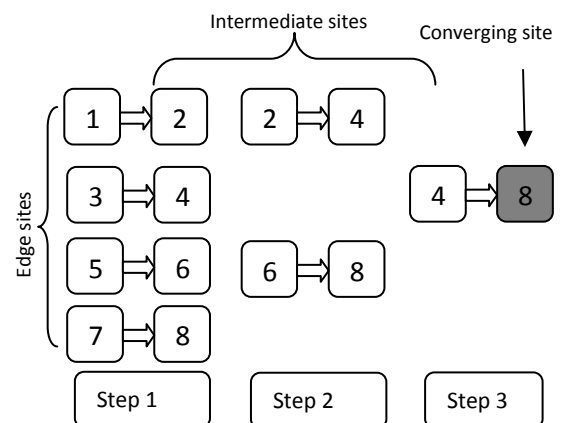


Fig.1: Depicts how distributed data is accumulated in a converging node (ReduceScatter).

Step 1: Nodes S_1, S_3, S_5 and S_7 transmit their counts to nodes S_2, S_4, S_6 and S_8 respectively.

Step 2: S_2 and S_6 transmit their counts along with the counts of S_1 and S_5 respectively.

Step 3: Now S_4 has counts of S_1, S_2 and S_4 , and then transmits all of them to node S_8 . Finally node S_8 will have the counts of all nodes. Now node S_8 is capable of calculating the sum of the counts.

Participating sites in the MPI technique can be divided into three categories: edge sites (in this example S_1, S_3, S_5 and S_7) are those which do not receive from others, intermediate sites (in this example S_2, S_4 and S_6) are those sites which receives from other sites and

converging site (in this example S_8) receives all information from others and accumulates.

In the second stage, the sum or unified value of all counts is transmitted back to all the nodes, which works in the reverse manner and known as AllGather.

In MPI; if there are N nodes in the network, then the total number of required transmission is $2(N-1)$, on contrary it is $N(N-1)$ in the case of broadcasting. The number of communication round is $2(\log_2^N)$.

Association Rule Mining: Let us consider; in a distributed data mining environment collective database DB is subdivided into DB_1, DB_2, \dots, DB_N in collection of data sites S_1, S_2, \dots, S_N respectively. $I = \{i_1, i_2, \dots, i_m\}$ is the set of items where each transaction $T \subseteq I$. Typical form of an association rule is $X \Rightarrow Y$, where $X \subseteq I, Y \subseteq I$ and $X \cap Y = \emptyset$. The support s of $X \Rightarrow Y$ is the probability of a transaction in DB containing both X and Y . On the other hand confidence c of $X \Rightarrow Y$ is the probability of a transaction containing X will contain Y too. Usually it is the interest of the data miners to find all association rules having support and confidence greater than or equal to minimum threshold value. Let us look at the equations of support and confidence for another instance of an association rule $AB \Rightarrow C$,

$$\text{Support}_{AB \Rightarrow C} = s = \frac{\sum_{i=1}^{\text{sites}} \text{support_count}_{ABC(i)}}{\sum_{i=1}^{\text{sites}} \text{database_size}_{(i)}}$$

$$\text{Support}_{AB} = \frac{\sum_{i=1}^{\text{sites}} \text{support_count}_{AB(i)}}{\sum_{i=1}^{\text{sites}} \text{database_size}_{(i)}}$$

$$\text{Confidence}_{AB \Rightarrow C} = c = \frac{\text{Support}_{AB \Rightarrow C}}{\text{Support}_{AB}}$$

More detail on association rule mining process is discussed by J. Han, M. Kamber (2006) and P. N. Tan, M. Steinbach, V. Kumar (2006).

Apriori algorithm proposed by R. Agrawal and R. Srikant (1994) is one of the leading algorithms, which determines all frequently large itemsets along with their support counts from a database efficiently. A brief description of the algorithm is as follows:

Let us say L_i be the frequent i -itemset. Apriori algorithm finds L_k from L_{k-1} in two stages: joining and pruning:

(i) Joining: Generates a set of k -itemsets C_k , known as candidate itemsets by joining L_{k-1} and other possible items in the database.

(ii) Pruning: Any $(k-1)$ -itemsets cannot be a subset of a frequent k -itemsets which is not frequent. Therefore it should be removed.

Count Distribution (CD): In brief, Count Distribution (CD) algorithm works as follows: Each processor or data site generates its local candidate sets based on the global large itemsets of previous iteration using Apriori algorithm. Then it calculates each support count and exchanges with other sites using Message Passing Interface (MPI) technique. Since this protocol exchanges all counts, each site can generate global frequent large itemsets which might be utilized for the following iterations. Due to the use of the same algorithm, all the processors generate same global frequent large itemsets. CD algorithm can be summarized into five major stages:

(i): Each processor generates candidate itemset C_k based on globally frequent large itemset L_{k-1} .

(ii): Each processor computes local support count for C_k by passing through the transactions in the database.

(iii): All processors exchange their C_k counts to develop global C_k using MPI technique.

(iv): Each processor computes L_k from C_k .

(v): Each processor takes the decision either to continue or to stop. Decision will be the same since they have identical L_k .

Fast Distributed Algorithm (FDM): Fast Distributed Mining of Association Rules (FDM), was proposed by W. Cheung, J. Han, V. T. Ng, A. W. Fu, Y. Fu (1996). The main idea of this protocol can be summarized as follows:

(i) Computing candidate set: Each site generates candidate set based on globally large $(k-1)$ -itemsets and locally large $(k-1)$ -itemsets using Apriori algorithm.

(ii) Local pruning: For each item in the candidate set: if the support of the itemset is larger than minimum support, that particular item is added in the locally large k -itemsets.

(iii) Count exchange: Each site broadcasts locally frequent large itemsets to all other sites.

(iv) Globally frequent large itemset computation: Each site computes globally large k -itemsets which is utilized for the following iteration.

3. Proposed Algorithm

Let us consider a distributed environment with N number of data sites $S_1, S_2 \dots$ and S_N possessing horizontally partitioned transactional data $DB_1, DB_2 \dots$ and DB_N respectively. All these sites intend to share their data to mine knowledge. Each site agrees to certain threshold values of minimum support (s) and confidence (c). This proposed algorithm generates all frequently large itemsets having their support values

more than or equal to s . Some of the notations used in the algorithm are included in the following table (Table 1):

Notations	Description
i-temset	An itemset consists of i elements/items
L_k	Set of all frequent k -itemsets sorted alphabetically. L_k is calculated and maintained in all the sites parallel.
C_k^i	Set of candidate k -itemsets in site S_i generated from L_k and sorted alphabetically.
λ_k^i	Set of count values of C_k^i in the order of C_k^i .
β_k^i	Set of calculated support values of C_k^i in the order of C_k^i .
V_k^i	Binary vector for C_k^i in the order of C_k^i . If $(\beta_k^i)_j \geq s$ then $(V_k^i)_j = 1$ else 0

Table 1: Notations

Each site maintains a table known as itemset-table to hold C_k^i , λ_k^i , β_k^i and V_k^i . A typical itemset-table is depicted in fig.3. In every iteration i , the table is updated based on the value of L_{i-1} in all sites.

Exchange of information is performed using MPI technique (discussed in previous section). Let us assume the communication sequence (who transmits to whom) and the converging node are predetermined. In the example provided in fig.1, S_5 sends its count to S_6 and S_6 will send counts of itself and of S_5 to S_8 . In that example S_8 is considered as the converging site. Therefore for N nodes S_N can be considered as the converging site.

A certain property of existence of itemsets in database is utilized in the proposed algorithm which is summarized as the following theorem: "If X is a globally frequent large itemset, there exists at least one site where X is locally large". Proof of the theorem is provided by W. Cheung, J. Han, V. T. Ng, A. W. Fu, Y. Fu (1996). Instead of exchanging the count of all itemsets, this algorithm transmits the information about whether a particular itemset is locally large or not (it is stored in the binary vector V_k^i) in the first attempt. In the following attempt sites only transmit the count of those itemsets which are locally large at least in one site. Thus a significant amount of communication overhead is reduced by avoiding transmitting unnecessary count values of all itemsets.

Major stages of the algorithm can be distinguished as follows:

Step 1: Each site S_i exchanges list of its items to all other $N-1$ sites using MPI technique. Each site computes list of 1-itemsets L_1 and sorts them in alphabetical order.

Step 2: Site S_i generates candidate set C_k^i from L_{k-1} by appending all non-repeated items with all itemsets in L_{k-1} . As for example let us say $L_1 = \{a, b, c, d\}$ and $L_{k-1} = \{ab, cd\}$ then $C_k^i = \text{sort}(\{abc, abd, cda, cdb\}) = \{abc, abd, acd, bcd\}$ where $k=3$. Then S_i computes its local λ_k^i , β_k^i and V_k^i and updates the itemset-table.

Step 3: If S_i is edge site (as depicted in fig.1), it transmits V_k^i to its pre-assigned sites. If S_i is intermediate sites, it receives all binary vectors from predetermined sites. Once all vectors from all expected sites are received, S_i performs OR operation among all vectors and transmits to its pre-assigned site. If S_i is converging site, it performs OR operation on all the vectors received and returns the resultant vector to all sites.

Let us assume an instance of a binary vector $[0 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1]$ which represents seven itemsets. The vector also implies that all itemsets are frequent except first and the fifth. As soon as this vector is received in the next node, it performs binary OR operation with its own binary vector and the received one. The result is transmitted to the following node (as illustrated in fig.2). It should be noted that all the nodes are transmitting the same amount of data which is the binary vector of fixed size for a particular iteration. If the number of itemsets are Q in a particular iteration, then the size of the vector would be \log_2^Q . At the final stage the converging node comes up with a resultant vector (let us name it V_r), which represents the frequent large itemsets of the following cycle (L_{i+1}). The figure fig.2 displays an example of a typical exchange of binary vector with some sites (S_1 , S_3 , S_5 and S_8):

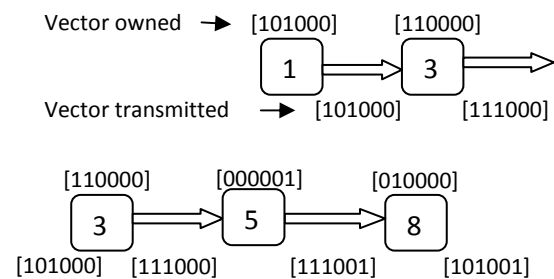


Fig.2: Computation of L_k from binary vector without exchanging the count values of itemsets.

In the above figure S_8 ; the converging site calculates $V_r = [111001]$ which, implies that first three and the last itemsets are frequent. Therefore $L_k = \{\text{first, second, third and sixth}\}$. Finally the converging site transmits V_r to all sites.

Step 4: Site S_i (for all $i \subseteq \{1, 2 \dots N\}$) removes the entries in the itemset-table for which the value in V_r is 0. Thus S_i generates L_k , which would be used to generate C_{k+1}^i in the following iteration. Thus the pruning (elimination of itemsets which have supports less than s) mechanism happens in step 3 and step 4 together.

Step 5: Site S_i exchanges C_k^i as the same way it was done in step 4 and 5 except addition operation is done instead of OR operation. In this case an integer vector would propagate instead of the binary vector. Therefore the size of the vector would be higher too. The converging site would receive the total count of all the itemsets which are at locally large at least in one site. Finally converging site returns the count values of all itemsets in L_k .

Step 6: Repeat step 2 to step 5 until there exists a large itemset with support $\geq s$.

Following figure (Fig.3) illustrates communication steps of the algorithm among data sites. Fig.3 also shows a typical entry of itemset-table of any site.

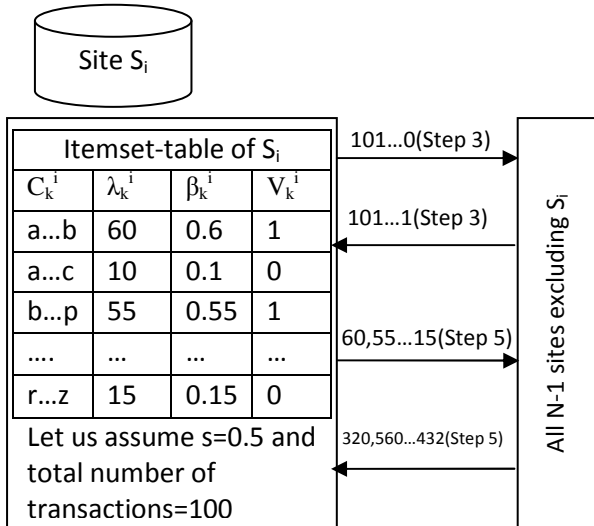


Fig.3: An instance of communication among data sites.

4. Performance Analysis

In this section a performance comparison would be presented to compare the number of communication rounds and amount of communication overhead necessary to transmit to compute frequent large

itemsets in every iteration. The analytical comparison involves CD, FDM and the proposed algorithm.

Let us consider following parameters:

H = Average number of items in the large k -itemsets (number of rows in the itemset-table).

L = Number of Bytes to store count values in itemset-table.

N = Number of data sites.

M = Average size of each item in Byte (number of characters as for example).

M' = Average number of items in each local candidate itemsets.

Communication overhead and number of communication round in each iteration for CD, FDM and proposed algorithm are as follows respectively:

CD algorithm: In CD; as discussed in the protocol description, data sites do not transmit the itemsets themselves. Instead, it transmits the counts of the itemsets since all sites have identical set of itemsets. Thus it reduces the amount of overhead to be transmitted in the network.

Therefore total communication overhead in each iteration P_{CD} = Overhead involved in forward communication (ReduceScatter) + Overhead involved in backward communication (AllGather) (as MPI technique discussed in Section 2)

$$P_{CD} = 2 * H * L * (N - 1)$$

It is also obvious from the protocol that, number of transmission is $O(N)$ and number of communication round is $O(\log_2^N)$.

FDM algorithm: In FDM local pruning reduces the number of items significantly. Therefore the overhead of FDM for each cycle is as follows:

$$P_{FDM-LP} = N(N - 1) * M * M' + N(N - 1) * 0.25 * H * L$$

It is claimed by W. Cheung, J. Han, V. T. Ng, A. W. Fu, Y. Fu (1996) that FDM reduces the number of itemsets by 75% to 90% by local pruning. According to the claim we consider their lower bound of optimization and assume average number of entries in the table be reduced to $0.25H$ in case of FDM. Therefore number of transmission is $O(N^2)$ and Number of communication round is $O(N)$.

The proposed algorithm: The proposed algorithm implements the pruning technique from the initial stages of the algorithm by avoiding transmitting the count of any itemset unless it is frequent at least in one site. Therefore pruning reduces the number of items significantly similar to FDM. For the sake of comparison it is assumed that the average number of entries in the table is reduced to $0.25H$; as it is in case of FDM. In that case, the overall overhead will be the summation of overhead in transmitting binary vector, overhead of returning resultant vector V_r from converging node and transmission of counts of frequent large itemsets. Therefore,

$$P = (N - 1) \times \left[\frac{2 \times H}{8} + 0.25 \times H \times L \right] \text{ which can be deduced to } P = (N - 1) \times H \times \frac{3L}{4}$$

Therefore number of transmission is $O(N)$ and number of communication round is $O(\log_2^N)$.

For the sake of performance comparison different parameters in the above derived performance equations are varied while keeping other parameters constant and assigning reasonable values to some of the parameters. Few performance comparisons are illustrated in Fig.4 and Fig.5.

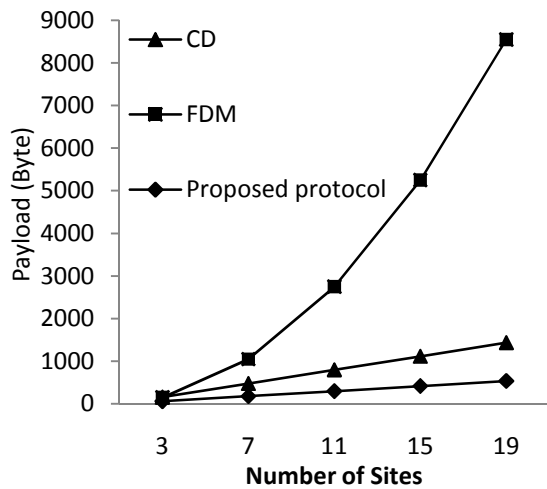


Fig.4: Average communication overhead in each cycle with H is 10, L is 4 M is 5 and M' is 3.

In fig.4 it is clearly depicted that as the number of data sites goes up, the overhead for FDM goes up most rapidly since it involves broadcasting. On the other hand CD has better performance since it does not broadcast to exchange the count information. On the other hand the proposed algorithm performs the best since it avoids broadcast and minimizes the size of the frequent large itemset. In addition to that, technique of binary vector exchange enables it to avoid exchanging the counts of non frequent itemsets. Therefore this algorithm minimizes the communication overhead most effectively.

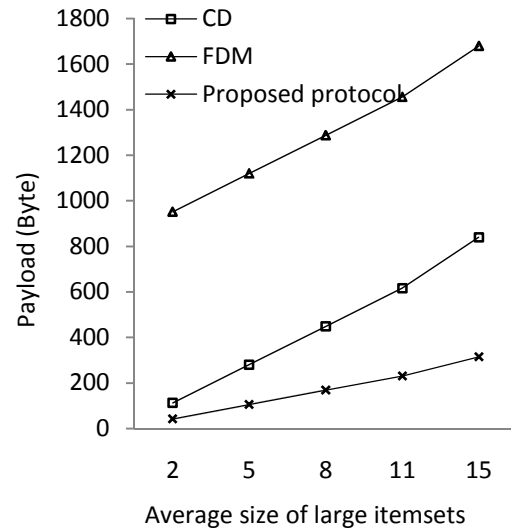


Fig.5: Communication overhead in each cycle with N is 8, L is 4 M is 5 and M' is 3.

Similarly fig.5 depicts that the proposed algorithm performs the best when average size of k-itemsets is varied keeping the number of sites constant.

5. Conclusion

FDM and CD have their own advantages and disadvantages in different circumstances and in various systems. Though FDM introduces some techniques to minimize candidate itemsets, it overloads the network by broadcasting too much data. On the other hand CD minimizes the communication overhead by avoiding broadcast but it does not consider optimization in frequent large itemsets generation. But avoidance of broadcasting, utilization of pruning technique and transmission of binary vector instead of count values reduces the network overhead significantly in the proposed algorithm. Performance equations show that the proposed algorithm performs better than CD and FDM in terms of communication overhead. Furthermore, centralized pruning and mining and binary vector exchange technique also would reduce some computational overhead in all sites, which is not considered in the performance evaluation. We believe further experiments and implementation of the proposed protocol would strengthen its efficiency and usefulness.

Acknowledgement

This research contribution was supported and funded by Australian Research Council (ARC) Discovery Project (DP0770479) – “Privacy Protection in Distributed Data Mining”.

References

- R. Agrawal and J.C. Shafer (1996): Parallel Mining of Association Rules”, Knowledge and Data Engineering, IEEE Transactions on Volume 8, Issue 6, pp. 962-969.
- W. Cheung, J. Han, V. T. Ng, A. W. Fu, Y. Fu (1996): A fast distributed algorithm for mining association rules, 4th International Conference on Parallel and Distributed Information Systems, 18-20 pp. 31-42.
- M. Kantarcioglu, C. Clifton (2004): Privacy-preserving distributed mining of association rules on horizontally partitioned data”, Knowledge and Data Engineering IEEE Transaction Volume 16, Issue 9, pp. 1026-1037.
- R. Agrawal and R. Srikant (1994): Fast algorithms for mining association rules, Proceedings of the 20th International Conference on Very Large Data Bases. Santiago, Chile: VLDB, pp. 487-499.
- R. Agrawal and J. Shafer (1996): Parallel Mining of Association Rules: Design, Implementation and Experience, Research Report RJ 10004, IBM Almaden Research Center, San Jose, Calif.
- Argonne National Laboratory (MPI): MPI web pages at Argonne National Laboratory URL: <http://www-unix.mcs.anl.gov/mpi>.
- Q. Ding, Q. Ding, W. Perrizo (2008): PARM—An Efficient Algorithm to Mine Association Rules From Spatial Data, IEEE transaction on systems, man, and cybernetics – part B: cybernetics, vol. 38, no. 6.
- K. Sun, F. Bai (2008): Mining Weighted Association Rules without Preassigned Weights, IEEE transactions on knowledge and data engineering, vol. 20, no. 4.
- F. Tao, F. Murtagh, M. Farid (2003): Weighted Association Rule Mining using Weighted Support and Significance Framework, SIGKDD,
- S. J. Yen, Y. S. Lee (2007): Mining High Utility Quantitative Association Rules, Springer Berlin, DOI: 10.1007/978-3-540-74553-2_26, pp. 283-292.
- J. Han, M. Kamber (2006): Data Mining Concepts and Techniques, Second Edition, Elsevier Inc. ISBN: 13:978-1-55860-901-3.
- P. N. Tan, M. Steinbach, V. Kumar (2006): “Introduction to Data Mining”, 1st Edition, ISBN/ISSN: 0321321367, 2006.

Applying Clustering and Ensemble Clustering Approaches to Phishing Profiling

John Yearwood, Dean Webb, Liping Ma, Peter Vamplew, Bahadorreza Ofoghi
and Andrei Kelarev

Internet Commerce Security Laboratory,
Center for Informatics and Applied Optimization.
University of Ballarat, Ballarat, Australia.

Email: {j.yearwood,d.webb,l.ma,p.vamplew,b.ofoghi,a.kelarev}@ballarat.edu.au

Abstract

This paper describes a novel approach to profiling phishing emails based on the combination of multiple independent clusterings of the email documents. Each clustering is motivated by a natural representation of the emails. A data set of 2048 phishing emails provided by a major Australian financial institution was pre-processed to extract features describing the textual content, hyperlinks and orthographic structure of the emails. Independent clusterings using different techniques were performed on each representation, and these clusterings were then ensembled using a variety of consensus functions. This paper concentrates on using several clustering approaches to determine the most likely number of phishing groups and explores ways in which individual and combined results relate. The approach suggests a number of phishing groups and the structure of the approach can aid the development of profiles based on the individual clusters. The actual profiling is not carried out in this paper.

Keywords: Clustering, Phishing, Graph Partitioning, Cluster ensembles, Profiling, Consensus functions.

1 Introduction

Phishing can be defined as a scam by which an email user is duped into revealing personal or confidential information which the scammer can use illicitly. Phishing attacks use both social engineering and technical subterfuge to steal personal identity data and financial account credentials. Phishing is one of the fastest growing scams on the Internet. The exclusive motivation of phishers is financial gain. Phishers employ a variety of different techniques from spoofed links to malware (keyloggers) to DNS Cache Poisoning (Stewart 2003) (which is also known as Pharming) to lure the unsuspected user into divulging their personal information (Emigh 2005).

A spoofed email is usually sent to a large group of people from an address that appears to be from their bank or some other legitimate institution. The phishing email is typically worded to instil a sense of urgency and to elicit an immediate response from the recipient, e.g., “verify your account details or your account will be closed”. The hoax email also contains a link to an online form that is branded to look exactly like the organization website. The form has to be filled in using sensitive information like passwords,

user account details and credit card details. Until recently most phishers used the names of financial institutions to deceive people into giving away their account information. They now also use the names of other organizations like eBay, PayPal and even the Australian Taxation Office.

Most technical approaches to phishing so far aim to detect and block or highlight phishing activities either in the original email or when the website is contacted. Examples include the work of Fette, Sadeh and Tomasic (2007), Wu et al (2006), Juels et al (2006), Chandrasekaran et al (2006), Chau (2005), and Jakobsson (2005). For example, the eBay Toolbar is a browser plugin that eBay offers to its customers, primarily to help them keep track of auction sites. The toolbar has a feature called Account Guard that monitors the domain names that users visit and provide warning in the form of a coloured tab on the toolbar. The tab is usually grey but it turns green if the user is on eBay or the PayPal site. It turns red if the user is on a site that is detected as spoofed by eBay. These approaches aim to protect an individual user from the actions of phishers, but they fail to address the issue of protecting the broader community.

Therefore here we propose the development of a complementary set of technology with the aim of profiling the behaviour of phishers, thereby allowing tracking, prediction and possibly identification of these illegal operators.

The rest of the paper is organized as follows: Section 2 gives the background of profiling. Section 3 provides the details of 3 groups of clusterings according to different feature types. Section 4 describes different types of consensus functions. Section 6 shows the evaluation methodologies, and experimental results are shown and discussed in Section 7. Finally Section 8 concludes the work and highlights a direction for the future research.

2 Profiling

‘Profiling is a data surveillance technique which is little understood and ill-documented, but increasingly used. It means generating suspects or prospects from within a large population, and involves inferring a set of characteristics of a particular class of person from past experience’ (Roger 1993). In (Roger 1993), different data surveillance techniques have been surveyed; like front-end verification and data matching. As well as different problems needing to be tackled in this area, it has been shown that profiling data requires different sets of measures. Take the definition of profiling as in (Roger 1993): ‘Profiling is a technique whereby a set of characteristics of a particular class of person is inferred from past experience, and data-holdings are then searched for individuals for close fit to that set of characteristics.’ Further-

more numerous potential areas for the use of profiling have been identified. These include patients who have a likelihood of suffering from certain diseases, students having potential artistic talents, identifying customers buying patterns and many others.

Forensic Psychology is used by Webb (2007) to identify perpetrator(s) of a crime based on the nature of the offence committed and its mode of operation (Alison et al. 2003), (Castle and Hensley 2002). This leads to determination of various aspects of criminal psychology before, during and after the crime is committed.

In this paper, we follow the same trend set up by these studies, to profile phishing emails based on their structural characteristics, their content and information about their likely origin. The approach in the first instance is to try to firmly identify the emails that are similar across all of these types of characteristics and assume that these correspond to different phishing groups with certain *modus operandi*. The next stage of the work (not reported in this paper), will be to construct profiles of these groups by identifying the link structure, orthographic structure and content character of each group.

3 Clustering techniques

The 2048 emails used in our experiments are a subset of a much larger corpus, obtained from a major Australian bank. These are emails gathered by their information security group over a span of 5 months in 2006, and were identified as phishing emails. Most of the emails are about 1026 characters in length and have both text and hyperlink content embedded in them. Some of them contain HTML script, including tables, images, links, and other structures that can be useful in differentiating the emails. Hence, defining the *modus operandi* of the individual phishing group or activity.

There are a number of features which could be used as a basis for comparison and clustering of these email documents. These include the actual text content displayed to the user, the textual structure of this content, the nature of the hyperlinks embedded within the message, or the use of HTML features such as images, tables and forms.

An approach is to represent each email document in terms of this set of features, and then apply a clustering algorithm to these feature vectors. However, there are two drawbacks to this simplistic approach. Firstly, the nature of the features is varied, some features are numeric whilst others are binary or categorical. Thus combining the features together in a single clustering algorithm is problematic. Secondly, clustering algorithms always produce a set of clusters, even if there is no evidence of any underlying structure in the data. In our case there are no ground-truth labels to use as a basis for testing the clustering results, as the actual source for any of the emails is unknown. Therefore, it is important that methods to validate the clusters produced by the system are found.

Similar issues have previously been observed in the context of clustering of high-dimensional data sets such as those used in bioinformatics. Researchers working in these areas have proposed the use of cluster ensembles. Several independent clusterings are performed based on different subsets of the complete feature vector. These are then combined together in a cluster ensemble to form a final, consensus clustering (Strehl and Ghosh 2002, Topchy et al. 2003, Fern and Brodley 2004). If pairs of objects (i.e. emails) are observed to be commonly grouped together across all of the independent clusterings, this provides increased

confidence that the clustering indicates a genuine relationship between the objects rather than just random noise. In (Fern and Brodley 2004) the different feature sets used in each clustering were random subselections or projections of the original feature vector. In our approach, we have instead chosen to use three groups of features which reflect different characteristics of the underlying data:

- the text content which is shown to the email's reader
- a characterisation of the hyperlinks in the email
- the orthographic features of the email

3.1 Text clustering

Perhaps the most obvious feature to use in profiling emails is the textual content displayed to the reader. For the emails in the data set this textual content was encoded in a number of ways. They included plain text, as HTML-formatted text, or as an embedded image. Therefore pre-processing was necessary to extract the text content from each email, by stripping away HTML tags and other structural information and by applying optical character recognition to the embedded images.

With the text extracted, this was then converted into a numerical feature vector for each email by computing the *TF/IDF* weight algorithm, shown in equation (1).

$$w_{ij} = tf \cdot idf = f_{ti} \cdot \frac{\log(N)}{df_k} \quad (1)$$

n = total number of emails

f_{ti} = frequency of term i in document k

df_k = number documents containing k

Each email was then represented as a vector of its term weights and these vectors were clustered using the k -means algorithm.

3.2 Hyperlink clustering

We grouped together emails based on similar tokens found within the fake hyperlink structure. Many of these hyperlinks contained similar names, this was especially apparent for their directory naming conventions. We looked for directory or file names that were obscure, frequently occurring and had names that were related to banking. We ensured that no legitimate directories or file names were included, as all legitimate bank hyperlinks were removed before the clustering process. All non legitimate and frequently occurring bank related names/tokens were used as a seed for each cluster. Any emails containing absent links, links with no directories, IP only or hex only links or links containing none of these key tokens were clustered together into the "other" cluster. Shown in Table 1 are the directory and file names used to build each cluster. The following is a more detailed description of the hyperlink clustering procedure.

3.2.1 Extraction of links

1. Firstly, we extracted all links from the 2048 emails.
2. We then pre-processed the links by removing all surrounding tags and script information and any other periphery not directly related to the file or name structure of the link itself.

3. Next we removed all legitimate links belonging to any bank.
4. The links were then broken up into their Protocol, Host name, multi level domains and multi level directory components. All protocol, host name and top level domain name identifiers were removed during this process. These include such things as “http:”, “www”, “edu”, “au” and all others.
5. All remaining tokens found in the second level domain and all other directory levels were then stored in a binary tree along with the number of emails they were found in and their overall corpus frequency count.
6. Any emails that had no tokens, due to absent links, legitimate bank links only or had no words present in the link, were automatically clustered into cluster 10, “Others” and didn’t contribute further to the clustering process.

3.2.2 Building the clusters

This was a partially manual process where we looked over the stored tokens, taking into account the names used, their overall frequency and the number of emails that they appeared in. With the possibility of such a large number of words, we ignored all words with a frequency of 1% of the total number of emails, and were not bank related. The exception however was the “moreinfo.html” and “wumoreinfo.html” file name. Words that may have occurred many times, but were found only in a few emails were also excluded as a grouping could not be formed from them. Large frequency words such as “index”, “netbank” or “bigpond”, were excluded as they were not unique enough to belong to just one group. However names such as “index2_files”, “nabib” and “.verify” were kept due to their higher obscurity and number of emails they were found in. Some examples are:

nabib

- <http://blog.co.tz/nabib/>
- <https://ib.national.com.au/nabib/help/>
- <http://startherefilms.com/nabib/>
- <http://evolk.info/nabib/>
- <http://floridanetservices.com/nabib/>
- <http://www.jr.ac.th/nabib/>

/r1/?

- <http://www.netbank.commbank.com.au.netbank.rim2s.biz/r1/c/>
- <http://citibusinessonline.da-us.citibank.com.dllinfo.tv/r1/cb/>
- <http://www.barclays.co.uk.customercare.goto.confpr.st/r1/b/>
- <http://www.national.com.au.vdq6270z.manicte.com/r1/n/>
- <http://www.barclays.co.uk.customercare.goto.mabberas.com/r1/b/>

Table 1: Number of emails in initial groups found

Hyperlink Keyword	Number of emails
/nabib	289
/.verifyacc/,/.ver/,/.verify/	22
/index2_files/	98
/anb2/	6
/r1/?/	765
/netbank/ or /netbank/bankmain/	319
cbusol	41
/.national.com.au	81
moreinfo.htm,wumoreinfo.htm	8
“Others”	419

With the seeding tokens now found, all remaining emails not belonging to the “Others” group were then allocated to the cluster containing their nominated token.

Shown in table 1 are the nominated fake link tokens (Cluster seeds) as well as the number of emails contained within each cluster. The “Others” row shows the remaining number of emails that didn’t contain viable hyperlink tokens.

3.3 Orthographic clustering

Phishing emails usually contain multimedia type information to help overcome phishing filters and lure the unsuspecting recipient. This includes images and text, where the text information may contain plain text, markup languages and styles, scripts, URLs and so on. The images may contain logos or a mock up of a bank or an institution’s web page with altered text. However, the information cannot be recognized by a system directly, rather it needs to be characterized according to the needs of the system.

Phishing emails are largely similar in content. Therefore, we believe that orthographic features are important in such an application. The orthographic features mainly consist of style characteristics that are used to convey the role of words, sentences or sections that describe the email content.

Since an email body is often loose in structure, parsing email content is more difficult than parsing the header part of the email. For the present we have defined the features manually based on observation. The orthographic features collected in our system are described as the following:

1. size of the text and html body of an email.
2. whether an email has text content¹.
3. number of visible links in an email.
4. whether a visual link is directed to the same hyperlink in an email.
5. whether an email contains a greeting line.
6. whether an email contains a signature at the end.
7. whether an email contains HTML content.
8. whether an email contains scripts.
9. whether an email contains tables.
10. number of images in an email.
11. number of hyperlinks in an email.

¹Some phishing emails contain only images

12. whether an email contains a form.
13. number of fake tags in a email².

A high level description of the feature extraction and clustering process can be seen in Figure 3. Features are collected according to features defined above, but not all the features are informative. Therefore, the most informative features are selected using a learning model and clustering is carried out. Both of these tasks are done iteratively using the Modified Global k -means algorithm (Bagirov and Mardaneh 2006, Bagirov 2008). A selection process conducts a search for a best feature subset and then uses Modified Global k -means (MGkm) for the evaluation of the current feature subset. This is run repeatedly on the phishing emails using various feature subsets and various tolerance values for MGkm. The performance is evaluated by MGkm objective function values on the various feature subsets, where the subset with the lowest objective function value is chosen as the iterated feature subset on which the induction algorithm runs.

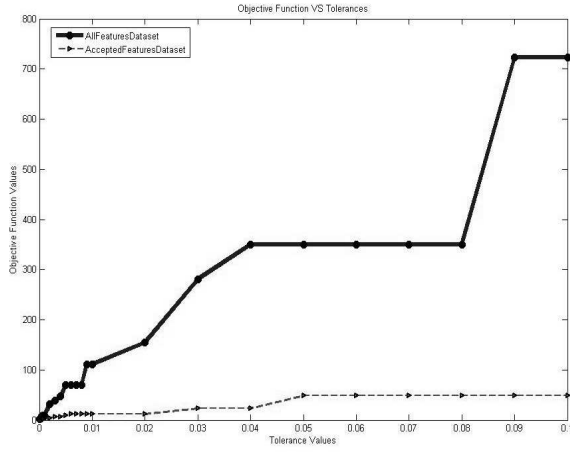


Figure 1: Objective function values vs tolerance values

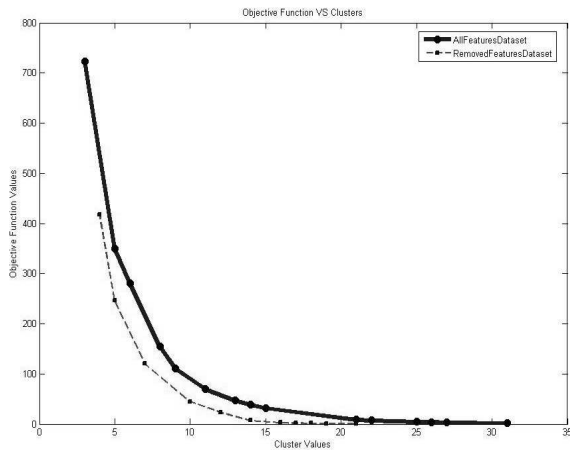


Figure 2: Objective function values vs the number of clusters

Figure 1 shows the relationship between the objective function values (V_{of}) and tolerance values (V_t)

²Sometimes phishers use ill formed HTML or embedded fake tags in an attempt to elude phishing filters

We calculated V_{of} over a range of V_t from 0 to 0.1 and discovered V_{of} achieves stable value of 45 when $V_t \geq 0.05$. Figure 2 illustrates the relationship between V_{of} and the number of clusters. The graph shows that when V_{of} is 45, the number of clusters is 9. Together these figures indicate that a good clustering (a good balance between objective function values, tolerance and the number of clusters) of this data set is achieved with 9 clusters.

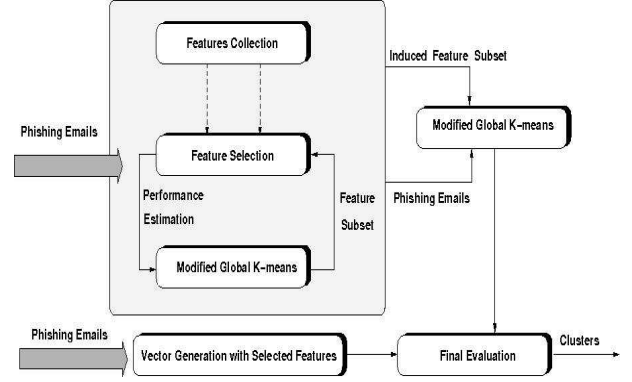


Figure 3: The feature selection and clustering process using orthographic features

4 Consensus functions

Several consensus functions have been proposed for forming consensus clusterings from an ensemble of independent clusterings (Strehl and Ghosh 2002, Topchy et al. 2003, Fern and Brodley 2004). Given a data set $X = \{x_1, x_2, \dots, x_n\}$ where n is the total number of instances x_l , $\Pi = \{\pi^1, \pi^2, \dots, \pi^r\}$ is a clustering ensemble on X where r is the total number of clusterings and $\pi^i = \{c_1^i, c_2^i, \dots, c_{K_i}^i\}$ where c_j^i corresponds to the cluster j in the clustering π^i and K_i is the total number of clusters formed in the clustering π^i . For each π^i we have $\cup_k c_k^i = X$.

Consensus clusterings are usually used on a single data set with different clusterings produced by:

- the different subsets of the whole feature set, or
- the different initial parameters in some clustering algorithms.

In this work, we use consensus functions on different clusterings obtained using the different features already discussed in Section 3. We utilize four consensus functions described by Fern and Brodley (2004).

- *Instance-Based Graph Formulation (IBGF)*: This method constructs a graph in which instances are represented by nodes and their connections are modelled as weighted edges given the association between the instances. The weight on the edge between the instances x_l and x_m in IBGF is calculated using the formula in equation (2).

$$w(l, m) = \frac{1}{r} \sum_{q=1}^r I(g_q(x_l) = g_q(x_m))$$

$$I(arg) = 1; \text{ if } arg = true$$

$$I(arg) = 0; \text{ if } arg = false$$

$$g_q(arg) = c_k^i; \text{ where } arg \in c_k^i$$
(2)

IBGF makes use of a graph partitioning algorithm³ to partition the graph according to the edge weights. The final clustering includes clusters corresponding to each graph partition.

- *Cluster-Based Graph Formulation (CBGF)*: This method constructs a graph with the clusters as graph nodes and the similarity of the clusters as weights on the edges. The edge weight between two clusters c_i and c_j in CBGF is calculated using the Jaccard Measure in equation (3).

$$w(i, j) = \frac{|c_i \cap c_j|}{|c_i \cup c_j|} \quad (3)$$

A graph partitioning algorithm is then used to eliminate the lowest weighted edges, thereby ensuring that clusters which share a large number of instances will be grouped together in the final consensus clustering. Following partitioning, each instance is assigned to the final cluster in which it most commonly occurs.

- *Hybrid Bipartite Graph Formulation (HBGF)*: This method constructs a bipartite graph with two types of nodes, clusters and instances. There is an edge between each pair of nodes; however, the weights of the edges between the nodes of the same type are 0. The edge weight between an instance x_l and a cluster c_i is 1 if $x_l \in c_i$ and is 0 otherwise⁴. The graph partitioning algorithm partitions both clusters and instances simultaneously and the final clustering is formed according to the partitions of instances.
- *K-Means Clustering Function (KMCF)*: This method was first proposed by Topchy et al. (2003) and uses the standard K-Means clustering algorithm to produce the final clustering. KMCF first creates a set of new features for each clustering π^i . It adds K^i binary features to the new set of features. Each of the new features correspond to a cluster in π^i . The total number of new features is equal to the total number of the clusters in Π . For each instance $x_l \in X$, the feature corresponding to c_i is 1 if $x_l \in c_i$ and is 0 otherwise. The new features are standardize to have zero mean and then the standard K-Means is applied to the features to create the final clustering.

5 Background to the experiments

Described in Section 3 are the text, links and orthographic structural clustering techniques. Each technique individually assigned each instance of an email to a cluster or profile according to its clustering criteria and feature set. Hence, capturing specific but different aspects of the data, where a single clustering technique alone could not. Our aim therefore is to combine these clusterings together to

- reinforce the intersecting information
- include information not shared between the three techniques
- find the best fitting number of profiles.

³In our implementations, we utilize the METIS graph partitioning module developed by Karypis and Kumar (1998).

⁴We set the weights between same node types to 1 and if an instance belongs to a cluster we set the weight to 1000. This is because of the implementation limitations in METIS where weights cannot be 0.

Shown in Section 4 is an explanation of the four consensus functions CBGF, HBGF, IBGF and KMCF. In terms of their application most work to date has been done on cluster ensembling of large data sets. The main focus here has been in breaking large data sets into smaller subsets using such techniques as random projections or random sub-sampling. Then, clustering algorithms could work on the smaller more manageable subsets of the original data set. The consensus functions would then be used to recombine these multiple clusterings so that a final clustering could be found. This work is shown in (Strehl and Ghosh 2002, Fern and Brodley 2004), where a comparison of all the consensus functions is undertaken on multiple data sets. Their results show that on average the HBGF and IBGF consensus functions improved the most when increasing both the number of projections and projections within the ensembles. The other two consensus functions performed less favourably under these conditions. However, these results were also dependent on the data set, and both the CBGF and KMCF consensus functions performed better than the other two on certain data sets.

Our application of these consensus functions is very different. We use different feature sets and clustering algorithms to find the clusterings. We also use a much smaller number in our ensemble, using only three technique clusterings. In (Fern and Brodley 2004) consensus functions are being compared in the use of many applications. One such application is Robust Centralized Clustering (RCC). Here, they look at a fixed number of clusterings; they have access to the dataset's features and use ten different but diverse clustering algorithms. This application is very similar to ours. The authors also show that a CBGF type consensus function performed very well in such an application. Unfortunately the KMCF consensus function was not examined in this study.

The culmination of the clusterings by the three techniques into one final clustering via the four consensus functions leaves us with at least four final clusterings. However, the number of cluster labels given to the consensus functions from the individual clustering techniques could vary between any or all of them. For example, in our case both the text and link clustering techniques have ten cluster labels, whereas the structural orthographic technique has nine. The consensus functions do not automatically determine what final number of cluster labels is the most appropriate. This means that we must specify the number of cluster labels for the final consensus clustering results.

From our previous examination of the fake links represented in the emails, an approximation of ten profiles was found. Furthermore, results from the structural orthographic technique shown in Section 3 using the Global Modified k -means algorithm (Bagirov 2008) reports nine clusters as an optimal number of clusters. We assumed then, that around 10 clusters could best partition the data set in terms of the number of profiles. With that in mind we set about establishing this final approximate number of clusters. In doing so, we casted a wider net by giving the consensus functions a range of 5 to 15 clusterings. This would allow us to evaluate five final cluster configurations either side of our initial assumption.

6 Evaluation Criteria

Evaluating the best final consensus clustering from 5 to 15 was a non-trivial process. To give us an indication of the "most correct" final clustering we employed the use of three measures, these were, Normalized Mutual information (NMI), purity and the number of edge cuts. We compared each final consensus

clustering 5 to 15 to each of the individual technique clusterings by comparing their intersections and relative information using the NMI and purity measures. We also compared the number of edge cuts given by the consensus functions from each final clustering. We surmised that the best final clustering would have the following:

1. A relatively consistent NMI value close to 1 when comparing the three given individual clustering techniques against all of the final consensus clusterings.
2. A high but consistent two way purity value. That is, a value similar when comparing both the individual technique to the final clustering and vice versa. We again would expect to have a value close to 1, to show that there is a strong intersection between both the individual technique clustering and the final consensus clustering.
3. A relatively low number of edge cuts given the number of clusters. This value is compared to all other clusterings within its respective consensus function as well as within the other consensus functions.

6.1 Purity

Purity measures the quality of a clustering solution by determining the number of points in the intersection of allocated clusters and predetermined labelled classes.

Let k be the number of clusters found by a hard partitioning clustering algorithm in data set D . Let $|c_j|$ be the size of cluster c_j and $|c_j|_{class=i}$ be the number of points of class i assigned to cluster j . Then the purity of cluster j is given by

$$purity(c_j) = \frac{1}{|c_j|} \max_i (|c_j|_{class=i}) \quad (4)$$

The overall purity for cluster $c_j, j = 1, \dots, k$, is expressed as a weighted sum of the k individual purities

$$purity(c_j)_{tot} = \sum_{i=1}^k \frac{|c_j|}{|D|} purity(c_j) \quad (5)$$

However, the purity measure shown here is asymmetrical. Let $|class_i|$ be the size of class c_i and $|class_i|_{clust=j}$ be the number of points of cluster j assigned to class i . Find $purity(c_i)$ and $purity(c_i)_{tot}$. Then $purity(c_j)_{tot} \neq purity(c_i)_{tot}$ unless the points are symmetrically distributed between both the respective class i and cluster j .

It is therefore a relative measure that depends on the order in which the multi labelled set of instances are measured. Since we are the ones attempting to label these instances we assume that we do not have the actual classification labels. We therefore, take $purity(technique_{pj})_{tot}$ of clustering technique p where p is an integer mapped to each technique type "links", "text" and "structural orthographic" against final consensus function m where $m = 1, \dots, 15$, the number of clusters found by each respective consensus function. We then find $purity(consensus_{mi})_{tot}$ against each $technique_p$ and $purity(technique_{pj})_{tot}$ against $consensus_m$. Leaving us with two purity measures comparing a two way symmetric intersection between the respective final consensus and individual technique clustering. Allowing us to measure the difference between the two. Hence, the smaller the distance between the two purity measures, the better the intersection between the two clusterings.

6.2 Normalized Mutual information

Mutual Information is a symmetrical measure that takes into account both the intersection of the two sets of clusterings as well as quantifying the statistical information found in both distributions, see (Cover and Thomas 1991). Though it provides a good indication of the shared information between a pair of clusterings, it is desirable as with purity, to have a normalized version of Mutual Information with values between $[0 - 1]$.

Let X and Y be random variables described by the consensus function clusterings $\lambda^{(i)}$ and the technique clusterings $\lambda^{(j)}$ where $i = 1 \dots p$ and $j = 1 \dots m$, with $k^{(i)}$ and $k^{(j)}$ number of clusters respectively. Let $I(X, Y)$ denote mutual information between two random variables X and Y and $H(X)$ and $H(Y)$ denote the entropies of both variable X and Y respectively. In the literature several normalizations exist. We chose the version of NMI found in (Strehl and Ghosh 2002, Fern and Brodley 2004) as it has been shown to successfully measure consensus functions against various types of ensembles. It uses the geometric mean of $H(X)$ and $H(Y)$ to normalize the mutual information see (Strehl and Ghosh 2002, Fern and Brodley 2004) for a detailed description and a proof.

$$NMI(X, Y) = \frac{I(X, Y)}{\sqrt{H(X)H(Y)}} \quad (6)$$

The NMI measure gives a best result when the value is close to one. This happens when the intersection of both $\lambda^{(i)}$ and $\lambda^{(j)}$ is strong and both entropies $H(X)$ and $H(Y)$ have similar values. Thus, NMI is a very good measure as it shows how much information has been preserved and how closely the clusters overlap between the final consensus results compared to the individual technique clusters.

An average or maximum of NMI values were used in (Strehl and Ghosh 2002, Fern and Brodley 2004) across the cluster ensembles created by the random projections or sub-sampling when compared against the final consensus clusterings. Since we have only three techniques this evaluation would be of no advantage to us. Therefore we decided to compare each individual technique against all of the final consensus clusterings individually, using a balance measure between the two purities and NMI to guide us to the best consensus clustering.

6.3 Number of edge-cuts

The Metis Graph partitioning software is used to create the partitions for the consensus functions shown in Section 4. The algorithm used, found in (Karypis and Kumar 1998), computes a k -way partition of a graph by minimizing the number of edge-cuts subject to a number of vertex balancing constraints. The edge-cut value is the total number of edges being cut in order to obtain that final number of clusterings.

We use this measure by dividing the number of edge-cuts by the number final clusters given to the consensus functions. As we have a range 5 to 15 final clusterings, we would expect the number of edge-cuts/number of final clusters to decrease across the 16 clusterings. It is natural that when asking the consensus function algorithm to produce a larger number of final clusterings, it would then make more cuts in order to create more partitions. However, because the algorithm works on minimizing the number of cuts, a number of cuts that is much greater than the previous cluster's cut would indicate a stronger cohesion amongst that partition.

We are then looking for a value that is considerably smaller than the clusters around it, as this shows that the number of cuts has decreased significantly.

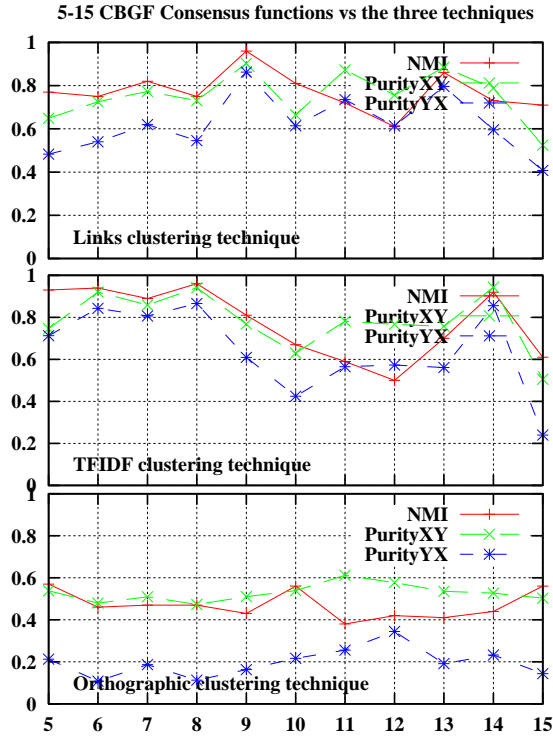


Figure 4: CBF Purity and NMI measures

7 Results

7.1 Comparing the individual techniques

We compared each individual clustering against one another to find their NMI, purity and individual entropy. As shown in Table 2, the three techniques contained roughly the same amount of information as they all displayed an entropy of between 0.62 – 0.74.

Table 3 shows the results of comparing the links and text content clusterings using both the NMI and purity measures. It can be seen that the links and text content clustering has the strongest intersection, as all the NMI and purity values are high. Leading us to the conclusion that both the links and text clustering techniques have captured similar information from the data set. However, Table 3 also shows that the structural orthographic clustering result compared to the other two techniques gave a much poorer NMI value. Furthermore, Table 3 also shows a big gap between the two orthographic purity measures, as well as these values being small. It is also worth noting that the results from comparing the Orthographic technique to the other two techniques were comparatively similar. This leads us to the conclusion that the structural orthographic technique has captured mostly different information compared to the other two techniques. This may also be the reason why, when comparing the three techniques to the final consensus clusterings that the Orthographic technique, again had poorer results compared to the other techniques.

5-15 IBGF Consensus functions vs the three techniques

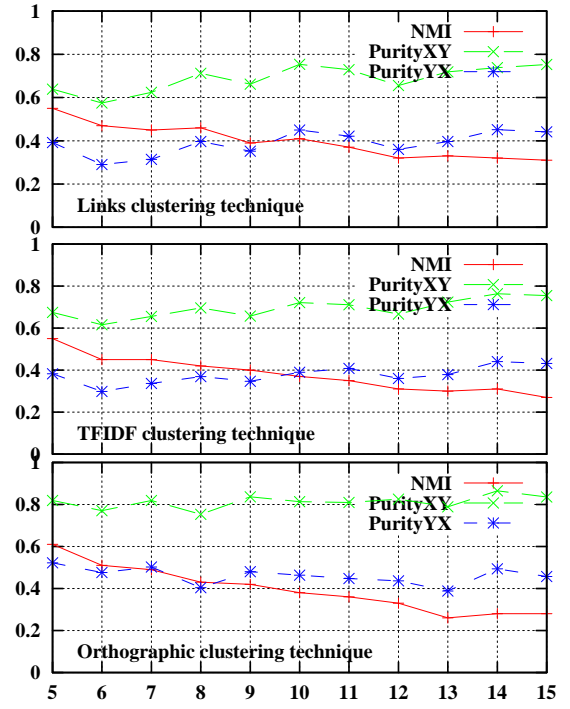


Figure 5: IBGF Purity and NMI measures

Table 2: Entropy of the individual clustering techniques

Technique	H(X)
links	0.737
Text content	0.674
Structural orthographic	0.615

7.2 Comparison of consensus functions

Figures 4, 5, 6 and 7 show the results of NMI and the purity values for each final consensus cluster 5 to 15 compared to each of the three techniques for the four consensus functions. When comparing the results shown in these graphs we can see a lot of variation in each of the different consensus functions. The CBF consensus function shown in Figure 4 and KMCF consensus function shown in Figure 7 report the best results. They show consistently higher NMI and purity values when compared to the IBGF consensus function shown in Figure 5 and HBGF consensus function shown in Figure 6. The other noticeable difference is that there is much less variation in the difference in the purity values of the CBF and KMCF graphs compared to the IBGF and HBGF graphs.

At closer inspection, we see that both CBF and

Table 3: NMI and purity results from comparing clusters techniques against one another

Technique	NMI	Pur(X,Y)	Pur(Y,X)
links vs text	0.77	0.7754	0.6104
links vs ortho	0.41	0.5103	0.1674
text vs ortho	0.46	0.4893	0.1268

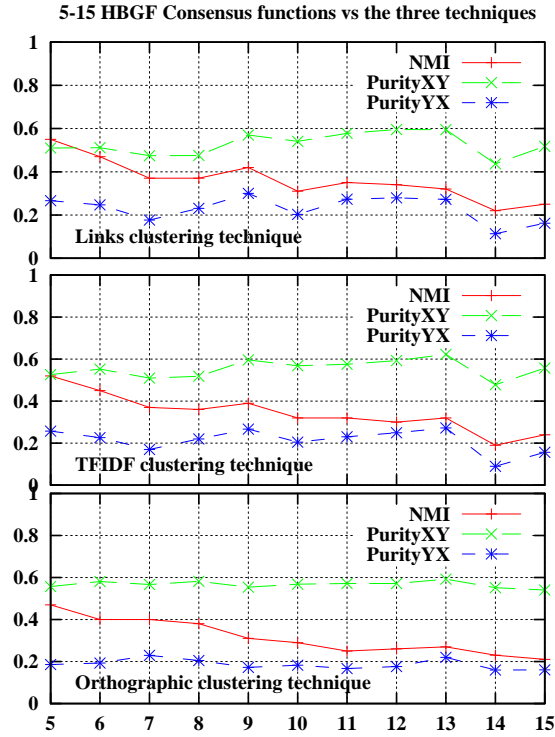


Figure 6: HBGF Purity and NMI measures

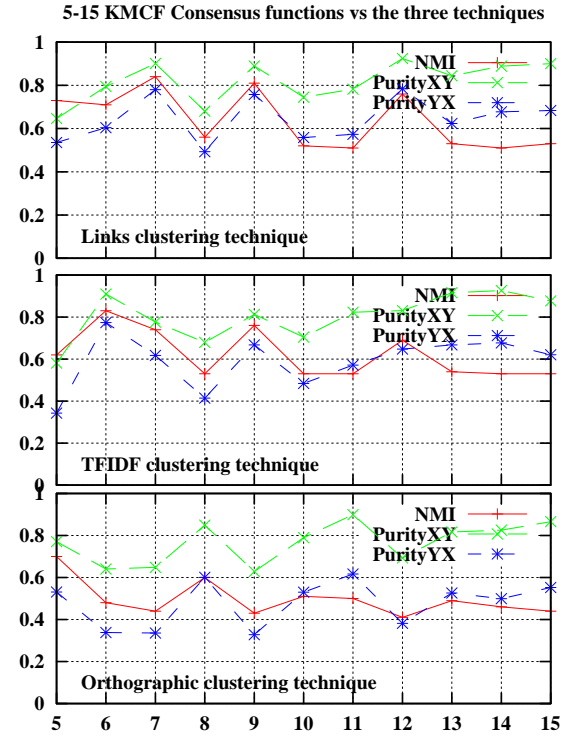


Figure 7: KMCF Purity and NMI measures

KMCF consensus function graphs show the highest NMI values for both the links and text content clustering techniques compared to results shown for the structural orthographic technique. Both the IBGF and HBGF consensus functions, shown in Figures 5 and 6 respectively report the worse results. Both consensus functions give the smallest NMI values compared to the CBGF and KMCF consensus function results. They also show that there are bigger differences in the two purity values again compared to the CBGF and KMCF consensus graph results.

Based on the results from these graphs we can rule out any of the final consensus clusterings produced by the IBGF and HBGF consensus functions. This leaves us with the clusterings obtained by the CBGF and KMCF consensus functions. When comparing the results of both the CBGF and KMCF consensus functions shown in Figures 4 and 7 respectively, we can see that the NMI values given in the CBGF consensus graph for both links and text/TFIDF techniques gives higher values at their respective peaks. The only exception is shown in the orthographic technique graph of the KMCF consensus function. The results shown for both the CBGF and KMCF consensus functions are favourable and warrant further exploring.

7.3 Evaluating the best final clustering

We can utilize both the NMI and purity measures to evaluate the best individual clustering. An ideal result for the final clustering would indicate a NMI value close to 1, both purity values would also be close to 1, but with a similar value. Balanced purity shown in equation (7) fulfils this criteria. It gives a value output in the range of 0 to 3 for each individual clustering technique, where 3 would be the best possible intersection and 0 the least. Our aim therefore, would be to find the largest balanced purity value

given across the three clustering techniques for each of the 5 to 15 final consensus clusterings of each consensus function.

We take the sum over the three clustering techniques for each of the 5 to 15 individual clusterings respectively. We then find the maximum value across all of the 5 to 15 clusterings for all four consensus functions, in order to find the overall maximum value. This maximum value would then give the best number of clusters for the best consensus function technique that would in theory best capture our data set.

We denote $purity(c_i)_{tot}$ as $pur(c_i)$ and $purity(c_j)_{tot}$ as $pur(c_j)$, refer to equation (5).

$$\text{balanced purity} = (NMI - |pur(c_i) - pur(c_j)| + pur(c_i) + pur(c_j)) \quad (7)$$

Figure 9 and Figure 10 show the results of equation (7) on both the CBGF and KMCF consensus

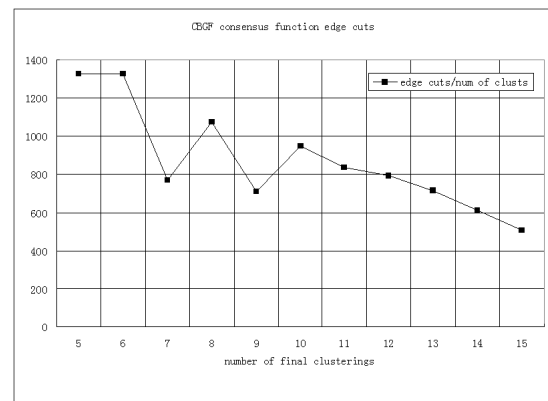


Figure 8: CBGF consensus function number of edge cuts/number of clusterings

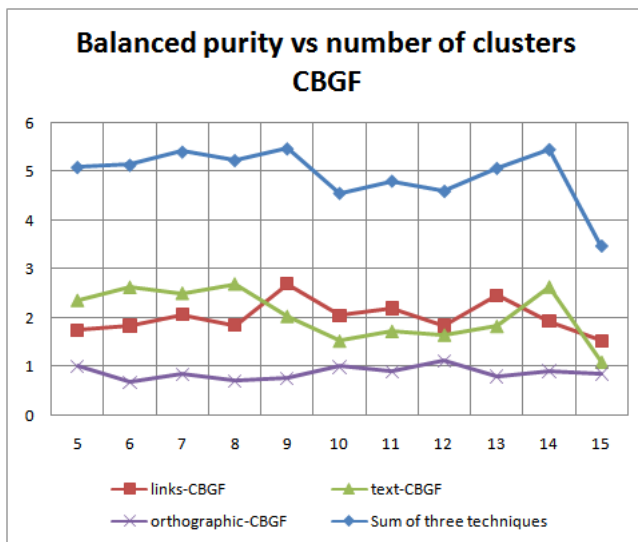


Figure 9: CBGF consensus function sum and individual balance purity measures for the links, text and orthographic techniques

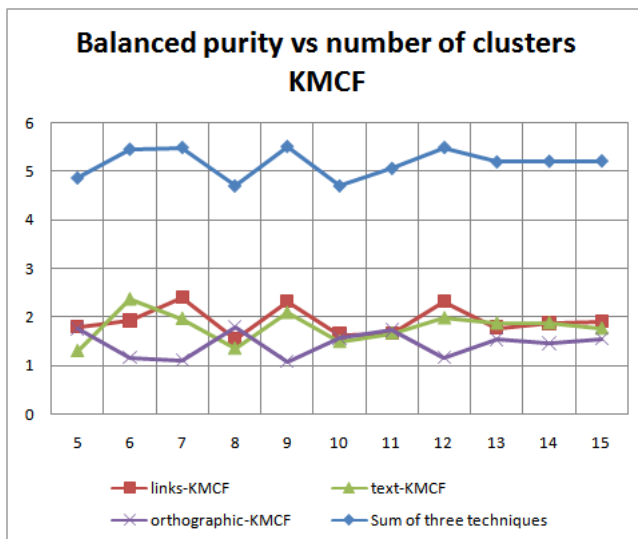


Figure 10: KMCF consensus function sum and individual balance purity measures for the links, text and orthographic techniques

functions against the three clustering techniques. The top values in the graph are the summation of the lower three sets of values that correspond to each of the individual clustering techniques. It can be seen, that in both of these graphs that the largest value found from the summation on the three techniques was the final clustering 9.

Figure 8 shows the number of cuts divided by the number of final consensus clusterings for the CBGF consensus function. As mentioned earlier, you would expect a decreasing graph with little fluctuations or pits in it. However, as shown in Figure 8 there are two significant dips, these are at final consensus clustering 7 and 9. This means that the number cuts for these two clusterings were considerably smaller than the cuts made in earlier and in later clusterings. This then leads us to the conclusion that both of these clusterings show the most stability in their partitions.

We can see from the results that the final clustering of 9 from the CBGF consensus function is the most consistent at gaining the highest values in terms of all our measurements. Though, other clusters have also presented comparatively good results, especially

within both the CBGF and KMCF consensus functions the clustering of 9 appeared to be the most consistent overall.

Finally, it is worth highlighting that the work undertaken in Section 3.3 found that a good clustering, (a good balance between objective function values, tolerances and the number of clusters) of the data set was achieved with 9 clusters. Refer to Figures 1 and 2.

8 Conclusion

Phishing is carried out by multiple groups of people over the internet. In this study we were provided with the artefacts of phishing attacks on financial institutions in Australia by a major Australian Bank. The artefacts of this phishing activity are emails that have been identified and classified as phishing emails. This work has used different clustering techniques to identify the groups involved in phishing. The main problems with emails is how to represent them as objects that can be clustered. Our approach has been to use three different representations of the emails, text content as determined by words, link content as determined by the hidden links in the email and the orthographic structure as determined by the features in Section 3.3. These were all natural representations of the emails, however a fourth facet of phishing emails is the scripting, but this will be part of our future work.

The features from each of the three feature spaces mentioned above were selected and individual clustering algorithms were used to determine clusterings based on each of these representations. Each of these clusterings provided different information, not all suggested a number of phishing groups. However the orthographic approach using the Modified Global k -means algorithm and some analysis of the objective function (clustering function) suggested 9 groups.

In order to utilise the evidence from the three clustering approaches, they were ensembled using the three clustering consensus approaches as described in Section 4. Two of these graphing functions, CBGF and KMCF provided interesting results when the edge cut graphs were examined, again suggesting 9 as the likely number of final clusters. The NMI and purity measures between these consensus functions and the three clusterings of the text, links and orthographic techniques also demonstrated maximum mutual information and balance purity around 9 clusters. This can be clearly seen by the sum in Figures 9 and 10.

Whilst not conclusive, this paper has explored clustering approaches and ensemble clustering approaches to provide information about the number of phishing groups. This, through the different individual clustering representations provides information about the profile of these groups. This paper has concentrated on assembling the strongest evidence for identifying a specific number phishing groups.

The issue of identifying and articulating the profile of these particular groups will be the subject of a further paper. A model will be built using the clusterings found in this paper, where the separate information about the modus operandi of the groups can be brought together. Other future work will include a reality check of our results with expert views of the number and nature of phishing groups and testing our model on other data sets.

References

- Alison, L., Smith, M., Eastman, O. & Rainbow, L. (2003), 'Toulmins philosophy of argument and its

- relevance to offender profiling', *Psychology, Crime and Law* **9**(2), pp. 173–183.
- Bagirov, A.M. & Mardaneh, K. (2006), Modified global k-means algorithm for clustering in gene expression data sets. In *Proc. 2006 Workshop on Intelligent Systems for Bioinformatics (WISB 2006)*, vol. 73, Hobart, Australia. CRPIT.
- Bagirov, A. M. (2008), Modified global k-means algorithm for minimum sum-of-squares clustering problems, *Pattern Recogn*, **41**, 10 (Oct. 2008), pp. 3192–3199.
- Bagirov, A.M. & Yearwood, J. (2006), A new nonsmooth optimization algorithm for minimum sum-of-squares clustering problems. *European Journal of Operational Research*, vol. 170 pp. 578–596.
- Castle, T. & Hensley, C. (2002), 'Serial killers with military experience: Applying learning theory to serial murder', *International Journal of Offender Therapy and Comparative Criminology* pp. 453–465.
- Chandrasekaran, M., Karayanan, K. & Upadhyaya, S. (2006), Towards phishing e-mail detection based on their structural properties, in *New York State Cyber Security Conference*.
- Chau, D. (2005), Prototyping a lightweight trust architecture to fight phishing, Technical report, MIT Computer Science And Artificial Intelligence Laboratory. Final Report, **URL**: <http://groups.csail.mit.edu/cis/crypto/projects/antiphishing/>
- Clark, R. (1993), Profiling: A hidden challenge to the regulation of data surveillance, *Journal of Law and Information Science* **4**, 2.
- Emigh, A. (2005), Online identity theft: Phishing technology, chokepoints and countermeasures, Technical report, Radix Labs. Retrieved from Anti-Phishing Working Group: **URL**: <http://www.antiphishing.org/resources.html>
- Fern, X. Z. & Brodley C. E. (2004), Cluster Ensembles for High Dimensional Clustering: An Empirical Study, *Journal of Machine Learning Research*.
- Fern, X. Z. & Brodley C. E. (2004), Solving cluster ensemble problems by bipartite graph partitioning, In *Proceedings of the Twenty-First international Conference on Machine Learning (Banff, Alberta, Canada, July 04 - 08, 2004)*. ICML '04, Vol. 69. ACM, New York, NY, p. 36.
- Fette, I., Sadeh, N. & Tomasic, A. (2007), Learning to detect phishing emails, in *WWW 07: Proceedings of the 16th international conference on WorldWide Web*, ACM Press, New York, NY, USA, pp. 649–656.
- Freund, Y. & Schapire, R. (1999), 'A short introduction to boosting', *Journal of Japanese Society for Artificial Intelligence* **14**(5).
- Thorsten, J. (2002), *Learning to classify text using support vector machines: methods, theory and algorithms*, Kluwer Academic Publishers, Norwell, MA, USA.
- Jakobsson, M. & Young, A. (2005), 'Distributed phishing attacks', *Cryptology ePrint Archive*, Report 2005/091. **URL**: <http://eprint.iacr.org/>
- Juels, A., Jakobsson M. & Jagatic, T. N. (2006), Cache cookies for browser authentication (extended abstract), in *SP 06: Proceedings of the 2006 IEEE Symposium on Security and Privacy (S&P06)*, IEEE Computer Society, Washington, DC, USA, pp. 301–305.
- Karypis, G. & Kumar V. (1998), A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices.
- Cover, T. M. & Thomas, J. A. (1991), *Elements of Information Theory*, Wiley, 1991.
- Fern, X. Z. & Brodley, C. E. (2004), 'Cluster ensembles for high dimensional clustering: An empirical study', *Journal of Machine Learning Research*.
- Karypis, G. & Kumar, V. (1998), METIS: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices, Technical report, University of Minnesota, Department of Computer Science and Engineering, Army HPC Research Centre, Minneapolis.
- Topchy A., Jain A. K. & Punch, W. (2003), Combining multiple weak clusterings, in *'IEEE International Conference on Data Mining'*, pp. 331–338.
- Strehl, A. & Ghosh, J. (2002), 'Cluster ensembles - A knowledge reuse framework for combining multiple partitions', *Journal of Machine Learning Research* **3**, pp. 583–617.
- Karypis, G. & Kumar, V. (1998), 'IEEE/ACM Conference on Supercomputing' SC98, 07-13 Nov. 1998 p. 28
- Roger, C. (1993), 'Profiling: A Hidden Challenge to the regulation of Data Surveillance', *Journal of Law and Information Science*, 1993 **4**(2)
- Webb, D. (2007), 'A Free and Comprehensive Guide to the World of Forensic Psychology', 'All about Forensic Psychology', **URL**: <http://www.all-about-forensic-psychology.com>
- Wu, M., Miller, R. C. & Garfinkel, S. L. (2006), 'Do security toolbars actually prevent phishing attacks?', in *'Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Montreal, Quebec, Canada, April 22 - 27, 2006)'*. CHI '06. ACM, NY, pp. 601–610.

Clustering Interval-valued Data Using an Overlapped Interval Divergence

Yongli Ren¹Yu-Hsn Liu²Jia Rong²Robert Dew²

¹ School of Information Engineering, Zhengzhou University,
Zhengzhou 450052, China
Email: yonglitom@gmail.com

² School of Information Technology, Deakin University,
221 Burwood Highway, Vic 3125, Australia
Email: {yuhsnliu, jrong, rad}@deakin.edu.au

Abstract

As a common problem in data clustering applications, how to identify a suitable proximity measure between data instances is still an open problem. Especially when interval-valued data is becoming more and more popular, it is expected to have a suitable distance for intervals. Existing distance measures only consider the lower and upper bounds of intervals, but overlook the overlapped area between intervals. In this paper, we introduce a novel proximity measure for intervals, called *Overlapped Interval Divergence (OLID)*, which extends the existing distances by considering the relationship between intervals and their overlapped “area”. Furthermore, the proposed *OLID* measure is also incorporated into different adaptive clustering frameworks. The experiment results show that the proposed *OLID* is more suitable for interval data than the Hausdorff distance and the city-block distance.

Keywords: Clustering, Distance, Similarity, Interval Valued Data.

1 Introduction

The importance of distance measures in machine learning and data mining is clear: a large number of learning problems, such as clustering and lazy learning, heavily rely on the similarity measurement over the data instance space. Accordingly, one of the main issues in these problems is the selection of a suitable metric for the concerned application domain. Most of existing distances have been designed for a relatively simple way: the data instance is described by a vector of random variables, each of which results in just one single value. However, in real life there are many situations where the use of interval-valued data is more suitable.

In general, interval-valued data come from two major sources:

1. many phenomena cannot be explained by using single-valued variables, and from their outset some data sets will include interval attributes. Many of natural language are expressed with intervals instead of single crisp values, e.g. “*I drink 4-6 cups of water a day.*” Similarly, in medical and engineering data, intervals also appear

Table 1: Sample Interval Data Set

No.	Age	Weight	...	Blood Pressure
1	[12, 17]	[45, 50]	...	[90, 100]
2	[25, 30]	[70, 80]	...	[138, 180]
...
21	[20, 30]	[65, 70]	...	[110, 150]
22	[10, 20]	[45, 70]	...	[90, 170]
23	[30, 40]	[70, 75]	...	[70, 120]

frequently, because of some tolerance in measuring real parameters. For example, age could be recorded as being in an interval, such as $[0, 10]$, $[30, 40]$ etc. In addition, it may not be possible to measure some characteristics accurately by a single value, e.g. the pulse rate at 70, but rather measures the variable as an $(x \pm \delta)$ value, namely (70 ± 1) . The blood pressure may be recorded by its $[low, high]$ values, e.g. $[138, 180]$. These are all interval-valued attributes. A typical data set with interval-valued attributes may follow the lines of Table 1.

2. As data sets increasingly suffer from the problem of scale, in terms of either the number of attributes or the number of instances. Researchers and practitioners from more diverse disciplines than ever before are attempting to use automated methods to analyze their data. It is often desirable to reduce the size of the data while maintaining their essential information as much as possible. One approach is to summarize large data sets in such a way that the resulting data set is of a manageable size. In this situation, interval data store variability better than standard single value data when real values describing the individual observations result in intervals in the description of the summarized data. Accordingly the summarized data could no longer be single values as in classical format, but instead be represented as intervals (Billard 2006).

The statistical treatment of interval-valued attributes has been considered in the context of *Symbolic Data Analysis* (SDA) (Diday 1988), which is a domain related to exploratory data analysis, multivariate analysis and pattern recognition. SDA aims to provide suitable methods for analyzing data set described through multi-valued attributes, including intervals, sets categories, or weight distributions. SDA has provided suitable tools for clustering interval-valued data: in 2004, Souza et al. proposed a clustering algorithm for interval data based on the *city-block* distance (de Souza & de A.T. de Carvalho 2004), and they applied the dynamic adaptive clus-

tering framework which incorporate the *city-block* distance to measure the distance between intervals. In 2006, De Carvalho et al. adopted a similar dynamic clustering framework but with the *Hausdorff* distance instead for intervals (de A.T. de Carvalho et al. 2006). Recently De Carvalho et al. further propose the single adaptive clustering framework, in which both the *city-block* and *Hausdorff* distances can be adopted (de A.T. De Carvalho & Lechevallier 2009). The single adaptive distances in (de A.T. De Carvalho & Lechevallier 2009) use the same adaptive parameters for all clusters, while it is different to the cluster adaptive distances in the early work (de A.T. de Carvalho et al. 2006), which use different adaptive parameters from cluster to cluster.

Most distances used for clustering interval data presented thus far have been designed for a relatively simple way: given two intervals, only the crisp values of their lower and upper bounds were considered, and the information about their overlapped area has been largely overlooked. However, in real life there are many situations where the ignorance of these overlapped area causes severe loss information, especially when both the distance between interval centers and the relative size of the overlapped area are concerned.

In this paper, we aim to fill the void by proposing a new distance for interval-valued data. By considering the intervals as a *hypercube* in a high dimensional space and take the overlapped area into consideration, the proposed *Overlapped Interval Divergence* is different from other interval distances which only consider their lower and upper bounds as single high dimensional points. As we will see from the later sections that by incorporating the proposed distances into both the single and the cluster adaptive clustering frameworks, we can get more accurate clustering results than existing distances.

The rest of the paper is organized as follows. In section 2, the related work are presented. In section 3, we propose the *Overlapped Interval Divergence* with detailed analysis of its properties and the adaptive clustering algorithms employed in the work. In section 4, we present the experiment results that evaluate the proposed algorithms compared to single(cluster) adaptive *Hausdorff* distance and single(cluster) adaptive *city-block* distance under the synthetic data sets. Finally conclusions and future work are presented in section 5.

2 Dynamic Clustering for Interval-valued Data

Clustering, partitioning data into sensible groupings according to measured or perceived intrinsic characteristics or similarity, is one of the most fundamental unsupervised data mining tasks. It is useful for helping user to understand and interpret the general patterns in data when prior knowledge of the underlying distribution is missing. As the representation of data by means of intervals is becoming more and more frequent, researchers and practitioners from more diverse disciplines than ever before are attempting to extend existing methods for the comparison of interval data (Diday 1988).

2.1 Interval-Valued Data

According to symbolic data analysis (Diday 1988), an interval variable is a variable which takes the interval values such as $[a, b]$, where $a \leq b$ and $a, b \in \mathbb{R}$. When $a = b$, this interval variable is becoming a normal single valued variable. Let D be a data set described by p interval variables. Each data instance $x_i \in D$ is

represented as a vector of intervals: $x_i = (x_i^1, \dots, x_i^p)$, where $x_i^j = [a_i^j, b_i^j]$.

A distance or proximity measure d is a non-negative function defined on each pair of interval-valued data instances, such that the closer the instances, the lower the value assumed by d . Two popular distance measures which have been widely used are the *city-block* distance (de Souza & de A.T. de Carvalho 2004) and the *Hausdorff* distance (de A.T. de Carvalho et al. 2006), which will be described later together with the dynamic clustering algorithms.

2.2 Adaptive Distances for Dynamic Clustering

Symbolic data analysis has provided clustering methods in which interval-valued data are considered. As the most influential symbolic data analysis method, the Dynamic Clustering Algorithm represents a group of unsupervised partition-based clustering algorithms. It can be proven that this group of algorithms generalizes several clustering algorithms including K -means and K -median algorithm.

The general Dynamic Clustering Algorithm looks for the partition of data set D into K clusters, and each cluster is represented by a single prototype vector of intervals, such that the sum of distance measures between each instance belonging to a cluster and the cluster's prototype is minimized.

Let $y_k = ([\alpha_k^1, \beta_k^1], \dots, [\alpha_k^p, \beta_k^p])$ be the prototype for the k -th cluster P_k ($k = 1, \dots, K$). The Dynamic Clustering Algorithm is then trying to minimize the following criterion:

$$O = \sum_{k=1}^K \sum_{x_i \in P_k} d_k(x_i, y_k). \quad (1)$$

Popular distance measures for interval-valued data include the *Hausdorff* distance (de A.T. de Carvalho et al. 2006) and the *city-block* distance (de Souza & de A.T. de Carvalho 2004). When they are used with Dynamic Clustering Algorithms, they usually appear in one of two adaptive forms: the single adaptive distance uses the same parameter for all clusters; the cluster adaptive distance uses different parameters from cluster to cluster (de A.T. De Carvalho & Lechevallier 2009).

2.2.1 The Single Adaptive Distances

Literature (de A.T. De Carvalho & Lechevallier 2009) proposes the partitional clustering algorithm for interval-valued data by using a single adaptive *Hausdorff* distance:

$$d_k(x_i, y_k) = \sum_{j=1}^p \lambda^j (\max[|a_i^j - \alpha_k^j|, |b_i^j - \beta_k^j|]), \quad (2)$$

in which $\lambda^j = (\lambda^1, \dots, \lambda^p)$ is a weight vector for p interval variables.

Similarly, if using the *city-block* distance we can get:

$$d_k(x_i, y_k) = \sum_{j=1}^p \lambda^j [|a_i^j - \alpha_k^j| + |b_i^j - \beta_k^j|], \quad (3)$$

where the weight vector $\lambda^j = (\lambda^1, \dots, \lambda^p)$ is also fixed for p interval variables.

De Carvalho et al. also propose an extended single adaptive *city-block* distance for interval-valued data clustering (de A.T. De Carvalho & Lechevallier 2009):

$$d_k(x_i, y_k) = \sum_{j=1}^p (\lambda_L^j |a_i^j - \alpha_k^j| + \lambda_U^j |b_i^j - \beta_k^j|), \quad (4)$$

in which there are two vectors of weight, one for the lower boundary $\lambda_L = (\lambda_L^1, \dots, \lambda_L^p)$, and the other for the upper boundary $\lambda_U = (\lambda_U^1, \dots, \lambda_U^p)$. These weight vectors are the same for each cluster.

2.2.2 The Cluster-Adaptive Distances

De Carvalho et al. introduce the dynamic clustering algorithm for interval data by adopting different adaptive *Hausdorff* distances for different clusters (de A.T. de Carvalho et al. 2006):

$$d_k(x_i, y_k) = \sum_{j=1}^p \lambda_k^j (\max[|a_i^j - \alpha_k^j|, |b_i^j - \beta_k^j|]), \quad (5)$$

which is parameterized by K vectors $\lambda_k = (\lambda_k^1, \dots, \lambda_k^p)$ ($k = 1, \dots, K$), one for each cluster.

Similarly, we can have the cluster adaptive *city-block* distance (de Souza & de A.T. de Carvalho 2004):

$$d_k(x_i, y_k) = \sum_{j=1}^p \lambda_k^j [|a_i^j - \alpha_k^j| + |b_i^j - \beta_k^j|], \quad (6)$$

which is also parameterized by K vectors $\lambda_k = (\lambda_k^1, \dots, \lambda_k^p)$ ($k = 1, \dots, K$).

Souza and De Carvalho also extended the cluster adaptive *city-block* distance by separately considering the lower and the upper bounds (de Souza & de A.T. de Carvalho 2004):

$$d_k(x_i, y_k) = \sum_{j=1}^p (\lambda_{kL}^j |a_i^j - \alpha_k^j| + \lambda_{kU}^j |b_i^j - \beta_k^j|), \quad (7)$$

where each cluster P_k is parameterized by two weight vectors: one for lower boundary $\lambda_{kL} = (\lambda_{kL}^1, \dots, \lambda_{kL}^p)$, the other for upper boundary $\lambda_{kU} = (\lambda_{kU}^1, \dots, \lambda_{kU}^p)$. Here, the weight vectors are also different from cluster to cluster.

2.2.3 The General Adaptive Clustering Algorithm

Once the strategy for adaptive distances is determined, the algorithm for clustering interval-valued data can be generated into a general process: it will randomly choose a partition of X into clusters $P = (P_1, \dots, P_K)$, then iterate over the following steps.

- In the first step, determine K cluster prototypes $y = (y_1, \dots, y_K)$ to represent each cluster.
- In the second step, fix the prototypes $y = (y_1, \dots, y_K)$ and the partitions $P = (P_1, \dots, P_K)$, and update the adaptive distances d_k so that the adequacy criterion O is minimized.
- In the third step, fix the prototypes and the adaptive distances, and determine the best partition $P = (P_1, \dots, P_K)$ which minimizes the adequacy criterion O .

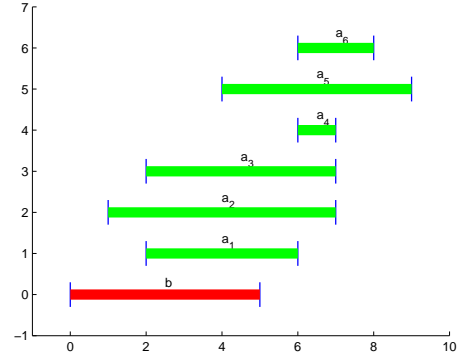


Figure 1: The interval $a_1, a_2, a_3, a_4, a_5, a_6$ and b .

3 Clustering with the Overlapped Interval Divergence

Most distances defined for interval-valued data have been designed for a relatively simple way: only the lower and upper bounds of the intervals are considered. However, in real life there are many situations where their overlapped area should also be considered (Li & Tong 2002, Li & Dai 2004, Jiang et al. 2005, Dai et al. 2004).

For example, considering seven intervals as shown in Fig. 1, the *city-block* and the *Hausdorff* distances from any interval $\{a_1, a_2, a_3, a_4, a_5, a_6\}$ to b are presented in Table 2. As we can see, intervals a_1, a_2, a_3 and a_5 are all overlapping with interval b , while intervals a_4 and a_6 have no overlapped area with b . According to the *city-block* distance, we have $d_c(a_1, b) = d_c(a_2, b) = 3$ and $d_c(a_4, b) = d_c(a_5, b) = 8$. It is evident that the *city-block* distance can not distinguish a_1 from a_2 , or distinguish a_4 from a_5 . Similarly, if following the *Hausdorff* distance, we will have $d_H(a_1, b) = d_H(a_2, b) = d_H(a_3, b) = 2$ and $d_H(a_4, b) = d_H(a_6, b) = 6$, which means the *Hausdorff* can not distinguish among a_1, a_2 and a_3 , or between a_4 and a_6 .

This is contradict to the intuition that a_1, a_2, a_3 are different from each other, especially when the relative size of the overlapped area is a concern. In this section, we are addressing this problem by proposing a new proximity measure for intervals.

3.1 The Overlapped Interval Divergence (OLID)

Any interval-valued data generalizes a single-valued data because it represents a range of values, and have “area” in nature. In addition, there will be an overlapped area between any two intervals, even though the overlapped area might be empty. For intervals, two factors are related to the proximity between two intervals: one is the distance between their centers; another one is the relative size of their overlapped area. By considering the above two factors together, we propose an *Overlapped Interval Divergence (OLID)* for intervals.

Definition 1 Given two intervals $a = [a_1, a_2]$ and $b = [b_1, b_2]$, let $c_a = \frac{a_1+a_2}{2}$, $r_a = \frac{a_2-a_1}{2}$ and $c_b = \frac{b_1+b_2}{2}$, $r_b = \frac{b_2-b_1}{2}$. Then the Overlapped Interval Divergence (OLID) from interval a to b is defined as:

$$\text{div}(a, b) = l(a, b) \cdot (1 - \frac{OA(a, b)}{2r_a + 1}). \quad (8)$$

Table 2: Distances to $b = [0, 5]$

Distances	$a_1 = [2, 6]$	$a_2 = [1, 7]$	$a_3 = [2, 7]$	$a_4 = [6, 7]$	$a_5 = [4, 9]$	$a_6 = [6, 8]$
city-block	3	3	4	8	8	9
Hausdorff	2	2	2	6	4	6
OLID	0.4	0.8571	1	3	3.3333	4

where $OA(a, b)$ is the Overlapped Area between a and b , and $l(a, b)$ is a distance originated from Hausdorff distance by considering all points inside the intervals:

$$l(a, b) = \max_{a' \in [a_1, a_2]} \{ \min_{b' \in [b_1, b_2]} \{ u(a', b') \} \}, \quad (9)$$

in which $u(a', b')$ is the Euclidean distance between a' and b' .

For an interval $a = [a_1, a_2]$, the relationship between a and any other interval $b = [b_1, b_2]$ could be divided into the following six types as shown in Fig. 2:

Falling Inside This kind of relationship occurs when interval a is completely falling inside of interval b , as shown in Fig. 2(a). In this situation we have the *OLID* $div(a, b) = 0$.

Covering This relationship happens when interval b is completely falling into interval a , as shown in Fig. 2(b). In this situation we have the *OLID* from a to b as $(|c_a - c_b| + r_a - r_b)(1 - \frac{2r_b}{2r_a+1})$.

Left Overlapping This corresponds to the situation where interval b overlaps with a on the left side of a , as shown in Fig. 2(c). We have $a_1 \in [b_1, b_2]$ and $a_2 \notin [b_1, b_2]$, then $div(a, b) = (|c_a - c_b| + r_a - r_b)(1 - \frac{r_a+r_b-|c_a-c_b|}{2r_a+1})$.

Right Overlapping This corresponds to the situation where interval b overlaps with a on the right side of a , as shown in Fig. 2(d). We have $a_1 \notin [b_1, b_2]$ and $a_2 \in [b_1, b_2]$, then $div(a, b) = (|c_a - c_b| + r_a - r_b)(1 - \frac{r_a+r_b-|c_a-c_b|}{2r_a+1})$.

Left Neighboring This happens when interval b is not overlapping with a , and b is on the left side of a , as shown in Fig. 2(e).

Right Neighboring This happens when interval b is not overlapping with a , and b is on the right side of a , as shown in Fig. 2(f).

By considering all the types of situations, we can get the *Overlapped Interval Divergence* function as follows:

$$\begin{aligned} div(a, b) &= l(a, b) \cdot (1 - \frac{OA(a, b)}{2r_a+1}) \\ &= \begin{cases} 0 & i \\ (|c_a - c_b| + r_a - r_b)(1 - \frac{2r_b}{2r_a+1}) & ii \\ |c_a - c_b| & iii \\ (|c_a - c_b| + r_a - r_b)(1 - \frac{r_a+r_b-|c_a-c_b|}{2r_a+1}) & iv \\ (|c_a - c_b| + r_a - r_b)(1 + \frac{|c_a-c_b|-(r_a+r_b)}{2r_a+1}) & v \end{cases} \quad (10) \end{aligned}$$

in which, i denotes when $|c_a - c_b| \leq r_b - r_a$; ii denotes when $|c_a - c_b| \leq r_a - r_b$; iii denotes when $r_a = r_b = 0$; iv denotes when $|r_a - r_b| < |c_a - c_b| < r_a + r_b$; v denotes when $|c_a - c_b| \geq r_a + r_b$.

It is interesting to note that when both intervals degrade into single values, the *OLID* divergence becomes the regular L_1 distance.

3.2 The Dynamic Clustering Algorithms based on Adaptive OLID

Based on the general framework of the adaptive clustering algorithm introduced in Section 2.2, the clustering algorithms based on single or cluster adaptive *OLID* are developed to discover the best partition of the original data sets into K clusters, which holds the minimum adequacy criterion O_{min} .

$$O_{single} = \sum_{k=1}^K \sum_{i \in P_k} d_{single}(x_i, y_k), \quad (11)$$

in which

$$d_{single}(x_i, y_k) = \sum_j^p [\lambda^j (\max\{div(a_i^j, \alpha_k^j), div(\alpha_k^j, a_i^j)\})] \quad (12)$$

is the *single adaptive OLID* measuring the dissimilarity between an object $x_i (i = 1, \dots, n)$ and a cluster prototype $y_k (k = 1, \dots, K)$, which is the median of $x \in P_k$ and multiplied by a weight vector $\lambda^j (j = 1, \dots, p)$. Here, since *OLID* is asymmetric, we use the *max* function to make it symmetric.

In each iteration, the weight vector $\lambda^j (j = 1, \dots, p)$ is calculated according to the following expression:

$$\lambda^j = \frac{\{\prod_{l=1}^p (\sum_{k=1}^K [\sum_{i \in P_k} (\max\{div(a_i^l, \alpha_k^l), div(\alpha_k^l, a_i^l)\})])\}^{\frac{1}{p}}}{\sum_{k=1}^K [\sum_{i \in P_k} (\max\{div(a_i^j, \alpha_k^j), div(\alpha_k^j, a_i^j)\})]}, \quad (13)$$

which satisfies $\lambda^j > 0$ and $\prod_{j=1}^p \lambda^j = 1$.

The Pseudo-code of the algorithms are presented in Alg. 1, which is both for single and cluster adaptive *OLID* algorithms.

For the dynamic clustering algorithm based on cluster adaptive distances share the same algorithm schema with the one based on single adaptive distances, but using a specific adequacy criterion $O_{cluster}$ since the adaptive distances are different from cluster to cluster:

$$O_{cluster} = \sum_{k=1}^K \sum_{i \in P_k} d_{cluster}(x_i, y_k), \quad (14)$$

in which the dissimilarity between the object x_i and the corresponding cluster prototype y_k can be calculated by the *cluster adaptive OLID* equation as follows:

$$d_{cluster}(x_i, y_k) = \sum_j^p [\lambda_k^j (\max\{div(a_i^j, \alpha_k^j), div(\alpha_k^j, a_i^j)\})]. \quad (15)$$

where the weight vector is

$$\lambda_k^j = \frac{\{\prod_{l=1}^p \sum_{i \in P_k} (\max\{div(a_i^l, \alpha_k^l), div(\alpha_k^l, a_i^l)\})\}^{\frac{1}{p}}}{\sum_{i \in P_k} (\max\{div(a_i^j, \alpha_k^j), div(\alpha_k^j, a_i^j)\})}, \quad (16)$$

which satisfies $\lambda_k^j > 0$ and $\prod_{j=1}^p \lambda_k^j = 1$.

4 Experiment and Discussion

To evaluate the performance of our *OLID* measurement, we investigate it within the framework of dynamic clustering algorithms on synthetic data sets.

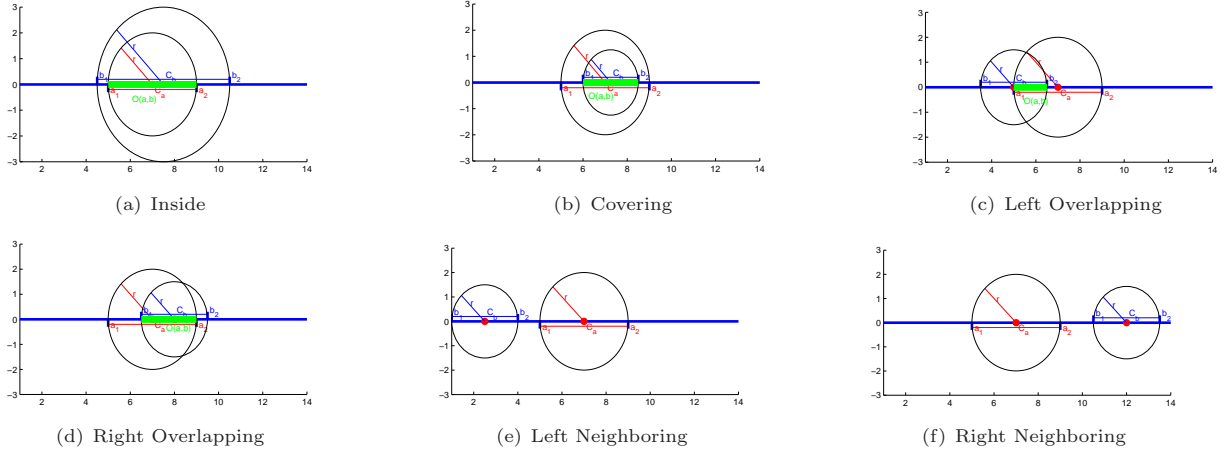


Figure 2: Different Relationships Between Two Intervals

Algorithm 1 The adaptive OLID algorithm**Input:**Data Set X .The number of clusters K .**Output:**A partition P of X into K clusters.**Algorithm Process:**1: **Initialization:**2: $P \leftarrow$ random partition of input data X into K clusters;3: **Iterative Research:**4: $Flag \leftarrow False$;5: while not $Flag$ 6: $Flag \leftarrow TRUE, change \leftarrow 0$;7: Calculate the prototypes $y_k (k = 1, \dots, K)$;8: Calculate the weight vector $\lambda^j (j = 1, \dots, p)$:

$$\lambda^j = \frac{\{\prod_{l=1}^p (\sum_{k=1}^K [\sum_{i \in P_k} (\max\{div(a_i^l, \alpha_k^l), div(\alpha_k^l, a_i^l)\})]\})\}^{\frac{1}{p}}}{\sum_{k=1}^K [\sum_{i \in P_k} (\max\{div(a_i^j, \alpha_k^j), div(\alpha_k^j, a_i^j)\})]}$$

or

$$\lambda_k^j = \frac{\{\prod_{l=1}^p \sum_{i \in P_k} (\max\{div(a_i^l, \alpha_k^l), div(\alpha_k^l, a_i^l)\})\}^{\frac{1}{p}}}{\sum_{i \in P_k} (\max\{div(a_i^j, \alpha_k^j), div(\alpha_k^j, a_i^j)\})},$$

9: for each element $x_i \in X$ 10: $k \leftarrow$ the label of the cluster which x_i belongs to;11: $k_{new} = \argmin_k (d_{single}(x_i, y_k))$ or $k_{new} = \argmin_k (d_{cluster}(x_i, y_k))$;12: if $k \neq k_{new}$ 13: Assign x_i to $P_{k_{new}}$;14: $change = change + 1$;

15: end if

16: end for

17: if $change = 0$, then $Flag \leftarrow TRUE$;

18: end while

for 100 times, and use the average CR over these 100 running for comparison.

4.1 Data Sets

In the experiments, three synthetic interval data sets are employed, which are designed to be well-separated, not-so-well-separated and over-lapping respectively.

4.1.1 Synthetic Data Sets

Following a similar strategy in (de Souza & de A.T. de Carvalho 2004, de A.T. de Carvalho et al. 2006, de A.T. De Carvalho & Lechevallier 2009), three types of the synthetic data sets are generated according to a bivariate normal distribution in a two-dimensional real number space, \mathbb{R}^2 . The first one represents a well-separated data set. The second one represents a not-so-well-separated data set, in which the class covariance matrices of the bivariate distribution are unequal; while for the third data set, the class covariance matrices are nearly the same. The parameters listed in Table 3 are set up to generate these three data sets respectively.

Each data set contains 450 data instances. A priori classification is done for evaluation convenience, by which four labels are set up to group the data instances into four classes with different sizes: *Class 1* and *Class 2* have the same size of 150, *Class 3* contains 50 data instances, and *Class 4* takes 100 ones.

As shown in Fig. 3(a), Fig. 3(c) and Fig. 3(e), each data instance (a_i, b_i) in well-separated, not-so-well-separated and over-lapping data sets is a seed of a vector of intervals:

$$([a_i - \gamma_1, a_i + \gamma_1], [b_i - \gamma_2, b_i + \gamma_2]),$$

where γ_1 and γ_2 are randomly picked up from intervals of $[1, 5]$, $[1, 10]$, $[1, 15]$ and $[1, 20]$. The data sets can be also represented by interval values as shown in Fig. 3(b), Fig. 3(d) and Fig. 3(f).

4.2 Clustering Validation

The Corrected Rand (CR) index, which was introduced in (Hubert & Arabie 1985), is one of the most popular clustering validation indexes (de Souza & de A.T. de Carvalho 2004, de A.T. de Carvalho et al. 2006, de A.T. De Carvalho & Lechevallier 2009).

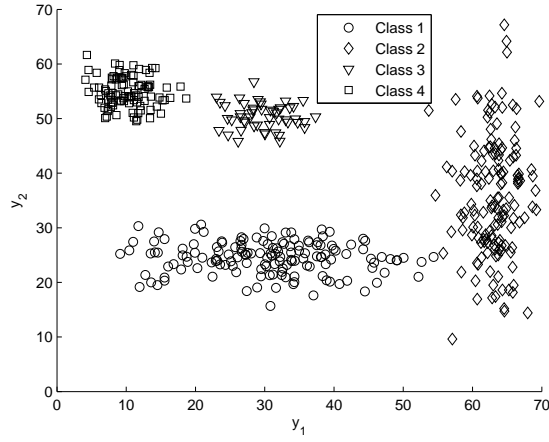
Given two partitions of the same data set, $U = \{u_1, u_2, \dots, u_R\}$ and $V = \{v_1, v_2, \dots, v_C\}$, which have

The results generated based on the *Hausdorff* and the *city-block* distances are also included for comparison purpose. This section starts with an introduction of the experimental data sets, then we describe the Corrected Rand Index (CR Index), which is widely used in the similar studies to evaluate the performance of the clusterings; finally, the experiments results of our *OLID* measurement are shown together with a discussion based on the performance comparison with the other popular measurements.

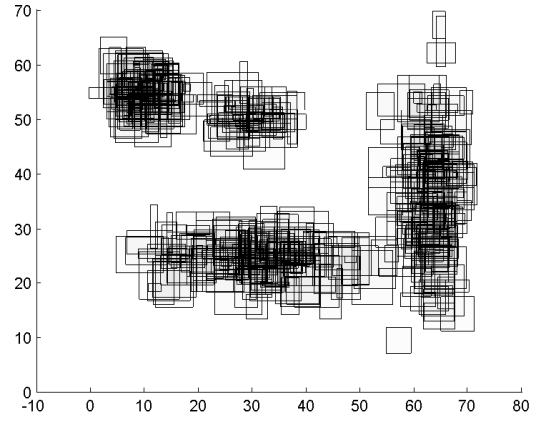
We compare our measurement with several popular distances, in the context of one/two weight single/cluster adaptive clustering algorithms. As a random initialization step in the dynamic clustering framework, we run each algorithm on each data set

Table 3: Parameters in the three synthetic seed data sets

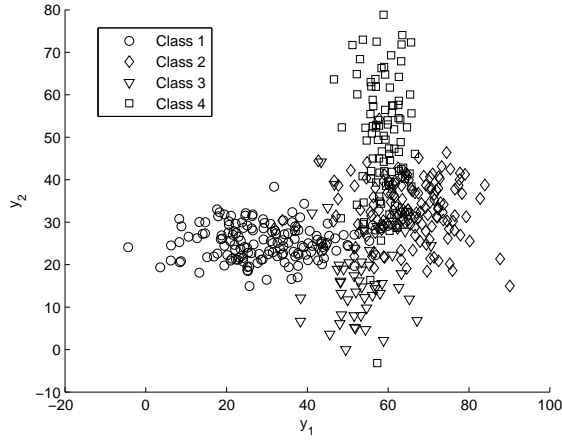
	Well-separated				Not so Well-separated				Over-lapping			
	μ_1	μ_2	σ_1	σ_2	μ_1	μ_2	σ_1	σ_2	μ_1	μ_2	σ_1	σ_2
Class 1	31	25	10	3	30	25	12	4	30	25	10	3
Class 2	63	34	3	12	64	33	9	7	64	32	9	4
Class 3	30	50	3	3	52	17	7	9	52	17	10	4
Class 4	10	55	3	3	59	50	4	14	59	39	9	3



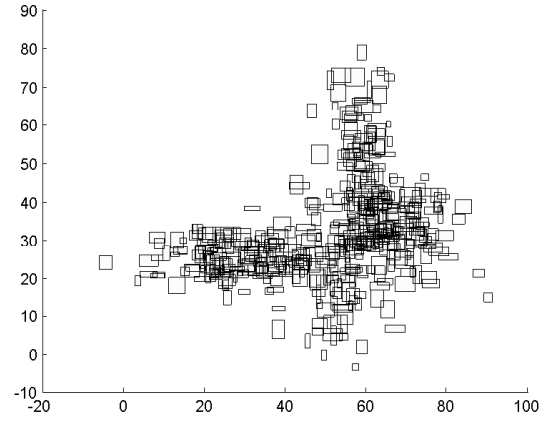
(a) Well-Separated Seeds



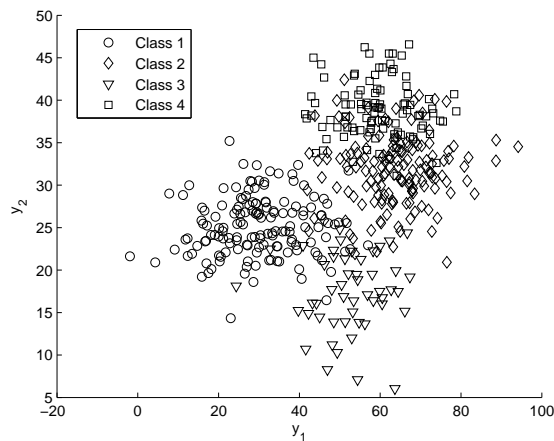
(b) Well-Separated Intervals



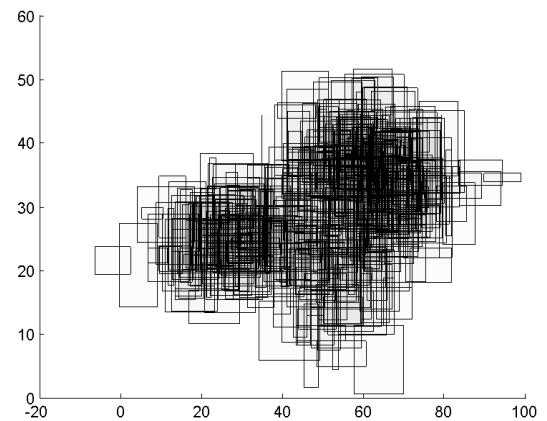
(c) Not so Well-Separated Seeds



(d) Not so Well-Separated Intervals



(e) Over-lapping Seeds



(f) Over-lapping Intervals

Figure 3: Three Synthetic Data Sets ($\gamma_1, \gamma_2 \in [1, 10]$)

R and C clusters respectively, the CR index can be estimated by the following equation:

$$CR = \frac{\sum_{i=1}^R \sum_{j=1}^C \binom{n_{ij}}{2} - \binom{n}{2}^{-1} \sum_{i=1}^R \binom{n_{i.}}{2} \sum_{j=1}^C \binom{n_{.j}}{2}}{\frac{1}{2} [\sum_{i=1}^R \binom{n_{i.}}{2} + \sum_{j=1}^C \binom{n_{.j}}{2}] - \binom{n}{2}^{-1} \sum_{i=1}^R \binom{n_{i.}}{2} \sum_{j=1}^C \binom{n_{.j}}{2}} \quad (17)$$

in which $\binom{n}{2} = \frac{n(n-1)}{2}$, n_{ij} represents the number of instances that are in clusters u_i and v_j ; $n_{i.}$ indicates the number of instances in cluster u_i ; $n_{.j}$ indicates the number of instances in cluster v_j ; and n is the

total number of instances in the data set.

The value of CR index for a certain clustering algorithm falls into the range of $[-1, 1]$. A CR index value of 1 indicates two clustering results are exactly the same, whereas the value 0 or below indicates that the cluster agreement found by chance (Milligan 1996). When comparing the clustering result with the true clustering partition, the higher the CR index value is, the better the result is.

4.3 Results and Discussion

Each clustering distance is incorporated with the single and cluster adaptive clustering framework, then run 100 times before the averaged CR is calculated.

4.3.1 Results for Synthetic Data Sets

Table 4 presents the values of the average and standard deviation (in parenthesis) of the CR index for the well-separated data set. It is evident that the proposed *OLID* distance in the cluster framework can get the best results on all the testing data sets. This is consistent with De Carvalho's findings (de A.T. De Carvalho & Lechevallier 2009), which discovered that the cluster adaptive clustering framework performs well on the well-separated data sets.

Table 5 presents the values of the average and standard deviation (in parenthesis) of the CR index for the not-so-well-separated data set. This is also consistent with De Carvalho's findings (de A.T. De Carvalho & Lechevallier 2009), which discovered that the cluster adaptive clustering framework performs well on the not-so-well-separated data sets. It is interesting to note that the proposed *OLID* distance in this framework can always lead to the best results on all testing data sets.

Table 6 presents the values of the average and standard deviation (in parenthesis) of the CR index for the over-lapping data set. This is also consistent with De Carvalho's findings (de A.T. De Carvalho & Lechevallier 2009) that the single adaptive clustering framework performs well on the over-lapping data sets. It is as expected that the proposed *OLID* distance outperforms all the other distances in the single adaptive clustering framework on all data sets.

4.3.2 Paired t-test Results

The two-tailed, paired t-test with 95% confidence level has been used to evaluate *OLID* with *city-block* and Hausdorff distance under single and cluster frameworks. The results are presented in Table 7. From the table, we can see that in the cluster adaptive clustering framework, the proposed *OLID* measure significantly improves the existing *city-block* and Hausdorff distances. In the single adaptive clustering framework, the *OLID* measure performs significantly better than the Hausdorff distance, and it is also significantly better than the two weight *city-block* distance, though the difference between it and the one weight *city-block* distance is not significant.

5 Conclusion

The choice of a distance measurement is essential for the success of many machine learning and data mining tasks, such as clustering and lazy learning. The trend of data representation as the interval-valued data calls for more sophisticated methods to evaluate the distance or similarity between interval-valued data instances.

The work is motivated by the fact that, most existing distance measures for interval-valued data only considered the lower and upper bounds, and overlooked the relative size of their overlapped area. In this paper, we introduce a new distance measurement based on the Hausdorff distance and the relative size of the overlapped area. We show its properties, and use it into different dynamic clustering frameworks: the single adaptive *OLID* algorithm and the cluster adaptive *OLID* algorithm. Our experiment results indicate the significant improvement of the proposed *OLID* measure over existing distances. In addition, our results further confirm that the single adaptive clustering framework is suitable for the overlapping data sets, while the cluster adaptive clustering framework is suitable for the well-separated data sets (de A.T. De Carvalho & Lechevallier 2009).

References

- Billard, L. (2006), Symbolic data analysis: What is it?, in 'Proceedings of 17th Symposium on Computational Statistics (COMPSTAT'06)', Physica-Verlag HD, Rome, Italy, pp. 261–269.
- Dai, H., Li, G. & Zhou, Z.-H. (2004), Ensembling mml causal discovery, in 'Proceedings of The Eighth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'04)', pp. 260–271.
- de A.T. de Carvalho, F., de Souza, R. M., Chavent, M. & Lechevallier, Y. (2006), 'Adaptive hausdorff distances and dynamic clustering of symbolic interval data', *Pattern Recognition Letters* **27**(3), 167–179.
- de A.T. De Carvalho, F. & Lechevallier, Y. (2009), 'Partitional clustering algorithms for symbolic interval data based on single adaptive distances', *Pattern Recognition* **42**(7), 1223–1236.
- de Souza, R. M. & de A.T. de Carvalho, F. (2004), 'Clustering of interval data based on city-block distances', *Pattern Recognition Letters* **25**(3), 353–365.
- Diday, E. (1988), *Classification methods of data analysis*, Elsevier, North Holland, Amsterdam, chapter The symbolic approach in clustering, related methods of data analysis, pp. 673–684.
- Hubert, L. & Arabie, P. (1985), 'Comparing partitions', *Journal of Classification* **2**(1), 193–218.
- Jiang, Y., Ling, J., Li, G., Dai, H. & Zhou, Z.-H. (2005), Dependency bagging, in 'Proceedings of The Tenth International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing (RSFDGrC 2005)', pp. 491–500.
- Li, G. & Dai, H. (2004), 'What will affect software reuse: A causal model analysis', *International Journal of Software Engineering and Knowledge Engineering* **14**(3), 351–364.
- Li, G. & Tong, F. (2002), 'Unsupervised discretization algorithm based on mixture probabilistic model', *Jisuanji Xuebao/Chinese Journal of Computers* **25**(2), 158–164.
- Milligan, G. (1996), *Clustering and Classification*, World Scientific, Singapore, chapter Clustering validation: results and implications for applied analysis, pp. 341–375.

Table 4: Well-Separated Data Set: comparison of the distances

Range of γ_i ($i = 1, 2$)	OLID		city-block				Hausdorff	
	One weight		One weight		Two weight		One weight	
	Single	Cluster	Single	Cluster	Single	Cluster	Single	Cluster
$\gamma_i \in [1, 5]$ CR Index	0.7464 (0.0935)	0.8102 (0.1038)	0.7393 (0.0879)	0.8005 (0.0954)	0.7378 (0.0926)	0.7875 (0.0967)	0.7228 (0.0726)	0.7859 (0.1071)
$\gamma_i \in [1, 10]$ CR Index	0.761 (0.1108)	0.8003 (0.0895)	0.773 (0.1197)	0.796 (0.0821)	0.7556 (0.1096)	0.7869 (0.0768)	0.7489 (0.1158)	0.7932 (0.0998)
$\gamma_i \in [1, 15]$ CR Index	0.7777 (0.098)	0.8122 (0.1037)	0.7455 (0.1088)	0.7715 (0.0863)	0.7558 (0.1157)	0.7805 (0.0939)	0.77 (0.1349)	0.7932 (0.0962)
$\gamma_i \in [1, 20]$ CR Index	0.7352 (0.1007)	0.8038 (0.0993)	0.6994 (0.0743)	0.7442 (0.0699)	0.6982 (0.0734)	0.755 (0.0872)	0.7173 (0.0909)	0.7345 (0.0801)

Table 5: Not So Well-Separated Data Set: comparison of the distances

Range of γ_i ($i = 1, 2$)	OLID		city-block				Hausdorff	
	One weight		One weight		Two weight		One weight	
	Single	Cluster	Single	Cluster	Single	Cluster	Single	Cluster
$\gamma_i \in [1, 5]$ CR Index	0.5235 (0.0470)	0.5555 (0.0589)	0.4967 (0.0183)	0.5275 (0.0808)	0.4905 (0.0354)	0.5315 (0.0768)	0.4691 (0.0520)	0.5303 (0.0834)
$\gamma_i \in [1, 10]$ CR Index	0.5139 (0.0642)	0.5682 (0.0611)	0.5262 (0.0386)	0.5561 (0.0392)	0.5305 (0.0364)	0.5526 (0.0435)	0.4788 (0.0829)	0.5332 (0.0937)
$\gamma_i \in [1, 15]$ CR Index	0.4329 (0.0091)	0.4869 (0.0347)	0.4309 (0.0172)	0.4683 (0.0118)	0.4327 (0.0071)	0.4691 (0.0017)	0.4132 (0.0181)	0.4198 (0.0428)
$\gamma_i \in [1, 20]$ CR Index	0.4606 (0.0412)	0.5225 (0.0487)	0.4883 (0.0335)	0.4967 (0.0843)	0.4792 (0.0232)	0.4946 (0.0725)	0.3568 (0.0270)	0.3769 (0.0444)

Table 6: Over-Lapping Data Set: comparison of the distances

Range of γ_i ($i = 1, 2$)	OLID		city-block				Hausdorff	
	One weight		One weight		Two weight		One weight	
	Single	Cluster	Single	Cluster	Single	Cluster	Single	Cluster
$\gamma_i \in [1, 5]$ CR Index	0.6079 (0.114)	0.6058 (0.1241)	0.5964 (0.0999)	0.5679 (0.0977)	0.5855 (0.0993)	0.5953 (0.1014)	0.5777 (0.1137)	0.5619 (0.105)
$\gamma_i \in [1, 10]$ CR Index	0.5958 (0.0536)	0.5826 (0.0765)	0.5672 (0.0627)	0.5451 (0.081)	0.5689 (0.066)	0.5521 (0.0809)	0.5598 (0.0292)	0.5355 (0.0548)
$\gamma_i \in [1, 15]$ CR Index	0.5211 (0.0211)	0.5350 (0.0722)	0.5168 (0.0394)	0.5040 (0.0759)	0.5203 (0.0479)	0.5163 (0.0847)	0.4816 (0.0445)	0.4720 (0.0566)
$\gamma_i \in [1, 20]$ CR Index	0.6416 (0.0845)	0.5546 (0.0947)	0.5755 (0.0654)	0.5548 (0.0799)	0.5691 (0.0654)	0.5395 (0.0911)	0.5335 (0.0604)	0.5090 (0.0684)

Table 7: Paired t-test Results

Algorithms	Distance	p-value	t
Single OLID vs. Single city-block	One weight	0.0961	1.8198
	Two weight	0.0492	2.2103
Single OLID vs. Single Hausdorff	one weight	0.0013	4.2584
	One weight	0.0004	5.0638
Cluster OLID vs. Cluster city-block	Two weight	0.0000	7.5389
	one weight	0.0006	4.738
Cluster OLID vs. Cluster Hausdorff			
	one weight	0.0006	4.738

Reference Point Transformation for Visualisation

Cheng G. Weng

Josiah Poon

School of Information Technologies,
J12, University of Sydney,
Sydney, NSW, Australia 2006,
Email: {cheng, josiah}@it.usyd.edu.au

Abstract

Visualisation of multi-dimensional dataset can be very useful for data mining purposes. This paper describes a simple visualisation technique to reduce a high dimensional dataset into a 3D space. Our aim is to design a method that is simple, easy, computationally cost-effective and able to give a reasonable visualisation of the dataset.

The original dataset is projected to a lower dimensional space via geometric metrics, while the proximity of the original data points is approximately preserved. The idea behind our data transformation is the concept of triangulation, which is applied through the use of reference points. In our study, we compared our method with the Principal Component Analysis (PCA) and Random Projection (RP). The results suggest that: when compared with PCA, our method can deliver a comparable visualisation of the dataset at a lower cost; when compared with RP, our method yields better visualisations at a similar cost.

Keywords: Data visualisation, Dimensionality reduction, Random projection

1 Introduction

“A picture is worth a thousand words”, a proverb that well describes why visualisation can be a powerful tool for gaining deeper insight into difficult problems. A good visualisation should be able to convey a story of what is being visualised. Generally, different domains require different types of visualisation techniques, for instance, we use stock charts to show stock market data and maps to display physical locations. It seems that the human brain is accustomed to handling information in graphical forms. A good example which shows that graphs are better analytical tools would be to compare reading experimental results from a 100x100 table, as oppose to reading them from a scatter plot. For any non-trivial sets of experimental results, it is generally easier to spot trends in the scatter plot than in the table. So a good visualisation should help to organise complex information into easy to understand structures. In this paper, the domain that we will try to visualise is the vector datasets, i.e. a collection of well-structured and stationary dataset, where each example in the dataset is represented by n number of features.

In general, there are some questions that need to be considered in order to construct a good visualisa-

tion for any vector dataset. Below are some of the basic questions that will help us shape our method:

1. **Meaningfulness:** Can the user *interpret* the visualisation in ways that would help them *learn* more about the dataset?
2. **Interactive:** Does the visualisation need to provide a *feedback* function to interact with the user and give real-time responses?
3. **Computation:** How long will it take to generate the visualisation? Does it scale up well with more data, i.e. more examples, higher dimensionality, or both?
4. **Limitations:** What would be the scope of the visualisation? Can it work with different types of attributes, or maybe an arbitrary number of dimensions?

Based on these questions, we have developed a data transformation technique to allow users to visualise datasets in an interactive environment. This technique uses different reference points to position examples in the dataset, therefore, we refer to it as the *kRef* method. While it is not meant to be an accurate approach as compared with other dimensionality reduction methods, we recommend it as a good visualisation alternative for practical reasons: it scales up well to large datasets, it is cheap to compute, it is memory efficient, it can be computed in parallel, it has no dimensionality restrictions, it is quick to implement and, most importantly, it is able to preserve a reasonable proximity of the original data points in order to deliver a meaningful visualisation of datasets.

In the next section, we will provide some related works, this will then be followed by Section 3, which is the main content of the paper and it describes our technique in detail. In Section 4, we will demonstrate the results of comparing our technique with PCA and RP, and in Section 5 we will provide some discussions and future works.

2 Related Works

Visualisation is one of the hot topics in computer science and there are many techniques that can be used to visualise multi-dimensional datasets. One of the early works for visualising multi-dimensional datasets was the parallel coordinates method (Inselberg, 1996). Parallel coordinates provides a 2D representation of any given dataset. A simple illustration is provided in Figure 1. The advantage of this method is that users can visualise the pairwise relationship between different dimensions in a fairly pleasant manner, but the drawback is that it can be very difficult to visualise when the dataset has a high dimensionality. There is also a novel approach of using human faces to represent different examples in the dataset (Morris

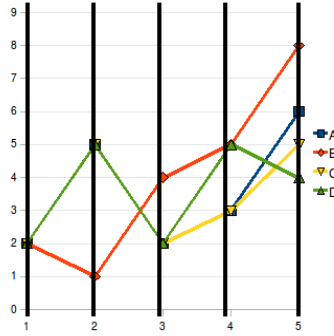


Figure 1: Parallel coordinates

An illustration of parallel coordinates. The x -axis is the dimensions/attributes and the y -axis shows the different values of the attributes. Examples are shown as lines connecting each dimension. In this figure, all four examples have the same value for Attribute 1, but they split into 2 groups on Attributes 2. In the last attribute, all four examples have a different value.

et al. (2000)). In this approach, multiple faces are presented, one face per example and each face have a number of adjustable facial features to represent different dimensions. For example, if one chooses to use the mouth to represent the first attribute, then the sizes of the mouth could be adjusted according to the value of the first attribute. In essence, this method makes use of the human's ability to spot similar and dissimilar faces, which translates to similar and dissimilar examples based on a combination of attributes. This method seems to be more scalable to dimensionality than the parallel coordinates method, but it is feasible only when dealing with a handful of examples.

Other related works in the data mining community are the dimensionality reduction techniques, such as PCA (Jolliffe (2002)), self-organising maps (Kohonen (2000)), random projection (Bingham and Manilla (2001); Lin and Gunopulos (2003); Blum (2006)) and multi-dimensional scaling (Borg and Groenen (1997)). For a good survey paper on dimensional reduction techniques, Fodor (2002) has provided a comprehensive coverage. Essentially, our technique also belongs to the category of dimensionality reduction, and the main challenge of dimensionality reduction is to perform feature selection with minimum information loss. The reduced dataset can then be utilised for other purposes, such as visualisation, data compression and efficient learning.

From an appropriate perspective, random projection (RP) is similar to our work because of the similar dataset transformation process. Random projection is based on Johnson-Lindenstrauss lemma (Johnson and Lindenstrauss (1984)), which states that if we project a set of m data points from n -dimensional space onto a random k -dimensional space, such that $k \geq O(\epsilon^{-2} \log(m))$, then the pairwise distances are preserved within ϵ . This minimum bound suggests that k could potentially be much smaller than n . To apply random projection, the original m by n data matrix is reduced, by multiplying a random n by k matrix, with the constraint that the column vectors in the random matrix are unit vectors. After the description of our work, in the discussion section, we will point out the differences between our method and random projection.

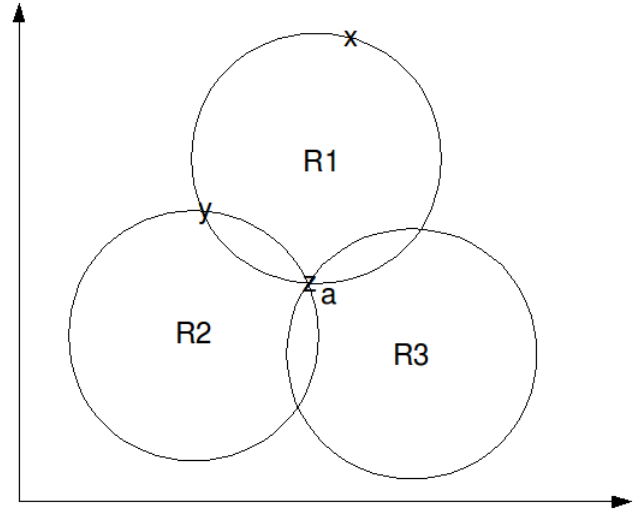


Figure 2: Reference Points

In this 2D example, we have 3 reference points (R1, R2 and R3) and 4 data points (x, y, z and a). The circles represent equal Euclidean distances from the centre reference points, i.e. the distances between R1 to x, y and z are the same.

3 Reference Point Method

As a motivational example of our proposed technique, we use Figure 2 to show that, using reference point R1 alone is not possible to tell apart x, y and z by the distance measurement, as they have the same distance. However, the situation can be improved with an additional reference point, R2, because x will have a different distance to R2 than y and z. Lastly, in order to separate y and z, we need another reference point, R3. Thus, with 3 reference points, we can uniquely identify x, y and z with their distance information to each reference point. Therefore, the basic requirement for the method to work is to have at least 3 reference points, and because we can select any k number of reference points, provided that $k \geq 3$, hence we called this method *kRef*.

Furthermore, we have put a data point, a, to show that, in the *3Ref* space, a and z should have similar distance measurements to each reference point because they are located at a close proximity; this is why *kRef* transformation is able to preserve the approximate pairwise distances in a lower dimensional space. One can see this transformation as projecting a dataset from the view of each reference point and combining different views to approximate the original image. However, the distances in the *kRef* space will be distorted due to the projection based on a distance metric.

3.1 Algorithm and Run-time

The *kRef* method will transform the original dataset, X , into a new space, $kRefTransform(X)$:

$$X = \begin{bmatrix} x_{00} & \dots & x_{0n} \\ \vdots & \ddots & \vdots \\ x_{m0} & \dots & x_{mn} \end{bmatrix}$$

$$kRefTransform(X) = \begin{bmatrix} kRef(x_0)_{00} & \dots & kRef(x_0)_{0k} \\ \vdots & \ddots & \vdots \\ kRef(x_m)_{m0} & \dots & kRef(x_m)_{mk} \end{bmatrix}$$

Algorithm 1 Algorithm for transform a dataset to *kRef* space (in Python).

```
def kRefTransform(dataset, referencePoints):
    newDataset = [] # create an empty dataset list
    for example in dataset:
        newExample = [] # create an empty example
        for ref in referencePoints:
            newExample.append(dist(ref, example))
        newDataset.append(newExample)

    return newDataset
```

The *kRefTransform()* method returns a new dataset by taking the original dataset, X , which contains m examples and n attributes; it then uses the function *kRef()* to map each example into k attributes, where k equals to the number of reference points. The basic algorithm is described in Algorithm 1, which is written in Python programming language. In Algorithm 1, we defined a method, *kRefTransform()*, that takes a list of examples (*dataset*) and a list of reference points (*referencePoints*), then returns with a list of examples in k -dimensions (*newDataset*). The *dist()* function in our implementation uses Euclidean distance, because we have found it to be a good choice in our experiments. Although the algorithm shows only one nested loop, but the Euclidean distance function actually contains another loop that goes through all the attributes in each *example*. So in the worst case, the algorithm has a run-time of $O(knm)$, where k is the number of reference points, n is the number of attributes and m is the number of examples in the dataset. In most cases, both k and n are fixed constants and only m would grow, so the average run-time is $O(m)$.

3.2 Fundamentals

We have identified 4 fundamental factors that can impact the resulting *kRef* space: the position of the reference points, the number of reference points used, the original dimensionality of the dataset and the distance metrics employed. These factors are inter-related, because different dataset dimensionalities and different number of reference points may require a different placement of reference points, which may in turn be affected by the distance metric. But, as a first attempt, we will analyse them independently.

Before we go through the factors, we will first propose a quality assessment measure for the dataset transformation.

3.2.1 Quality measurement

The measurement is intended to evaluate how good a technique is at preserving the proximity of data points in the geometric space. We define this measurement as the correlations between the pairwise distances in the original space and that in the transformed space. For efficiency reasons, if the dataset is large, we will only examine a random subset of the dataset; our empirical results suggest that a sample of 20% seems to give a reasonable estimate of the true value.

For a given set of data points, we need to first generate a matrix that contains $m \times m$ pairs of distances:

$$\begin{bmatrix} \text{dist}(x^{(0)}, x^{(0)}) & \cdots & \text{dist}(x^{(0)}, x^{(m)}) \\ \vdots & \ddots & \vdots \\ \text{dist}(x^{(m)}, x^{(0)}) & \cdots & \text{dist}(x^{(m)}, x^{(m)}) \end{bmatrix}$$

The *dist()* function calculates the Euclidean distance between two examples. This matrix is symmetric, i.e. the diagonal line contains zeros and the upper-right half is a mirrored copy of the lower-left half. Therefore, we can flatten the matrix into a vector by appending the rows together, i.e. $D^{\text{sample}} = [\text{row}(x^{(0)}), \text{row}(x^{(1)}), \dots, \text{row}(x^{(m-1)})]^T$, and the size of D^{sample} is $\mathbb{R}^{m(m-1)/2}$.

We compute the distances vector for both the original and the transformed datasets to get D^{original} and D^{kRef} , then *correlation*($D^{\text{original}}, D^{\text{kRef}}$) can be calculated with Pearson Product-moment Correlation Coefficient (Moore, 2006):

$$\text{correlation}(X, Y) = \frac{1}{M} \sum_{i=1}^M \left(\frac{X_i - \mu_X}{\sigma_X} \right) \left(\frac{Y_i - \mu_Y}{\sigma_Y} \right)$$

where μ_X , μ_Y , σ_X and σ_Y are the means and the standard deviations for the corresponding vectors, and M is the size of X , i.e. $|X|$. The function *correlation()* will produce a real number between -1 and 1 to indicate whether the vectors have a negative correlation (towards -1) or a positive correlation (towards 1), or simply no obvious correlation (around 0). A good quality data transformation should have a strong positive correlation, whereas a poor quality transformation is indicated by severe negative correlation, because it means that the original distances are distorted in the transformed space.

3.2.2 Synthetic test datasets

Test datasets are used to examine the effects of each factor on the quality of the transformed space. A test dataset consists of data points at every possible feature space, e.g. if a dataset is described by 3 attributes and each attribute can take on 5 different values, then all possible spaces in this dataset will be $5^3 = 125$. We have generated 64 test datasets with all combinations from different settings: the original dimensionality $\{2, 3, 4, 5\}$, the number of reference points $\{3, 4, 5, 6\}$ and the possible attribute values $\{3, 4, 5, 6\}$.

3.2.3 Factor 1: Positions of the reference points

It can be shown that the quality of the resulting *kRef* space will be affected by the placements of the reference points. Suppose, in Figure 2; instead of the stated 3 reference points, we choose y , z and a to be the reference points, then the new reference points will form a line. When this happens, the line is effectively a mirror-like projection, i.e. the data points on either side of this line will have a symmetrical counterpart that has the same distances to each new reference points. As a result of this mirror-like projection, two

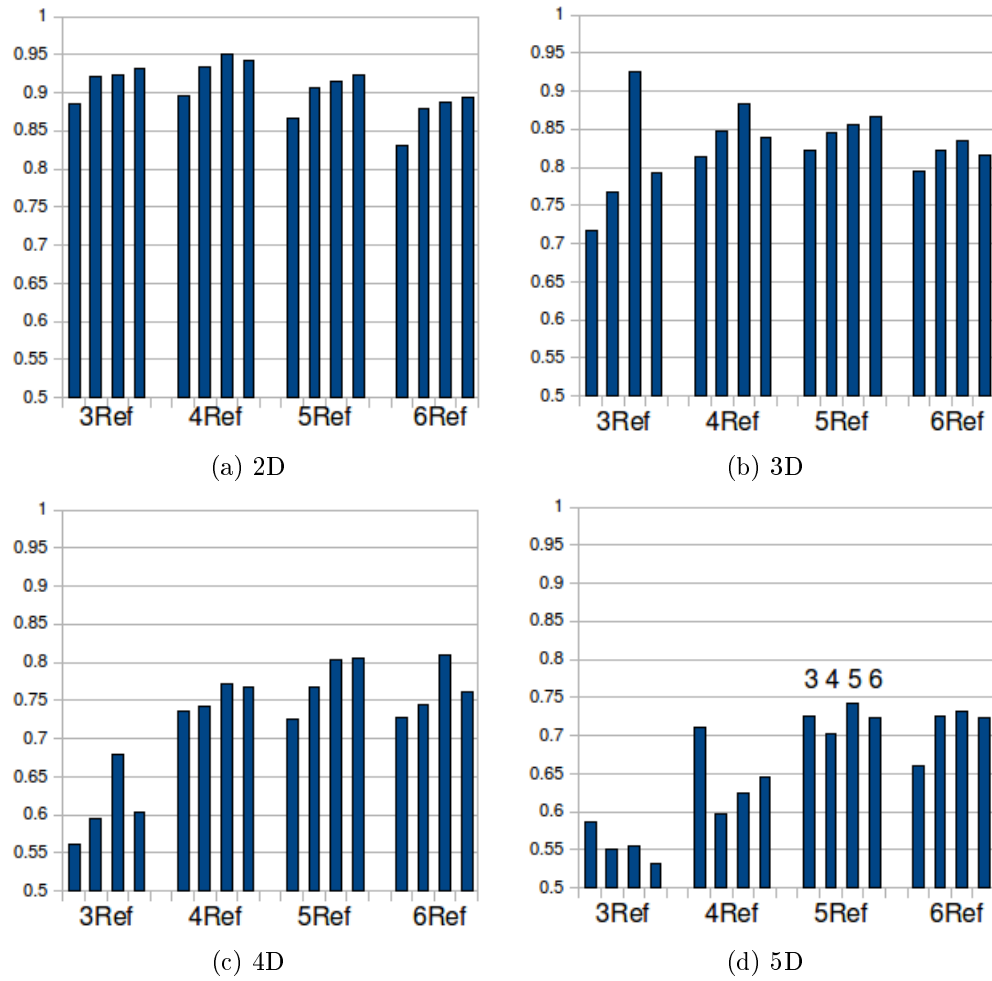


Figure 3: Use test datasets to compare different factors

The y -axis is the quality measurement described in Section 3.2, so large value means higher quality. These 4 graphs represent 4 different original dimensions (2D to 5D), and in each graph there are 4 different clusters representing different numbers of reference points used ($3Ref$ to $6Ref$), then each cluster is further divided into 4 bars, where each bar, from left to right, represents the number of possible attribute values (3 to 6). For visual consistency, the graphs are scaled to sit between 0.5 and 1.

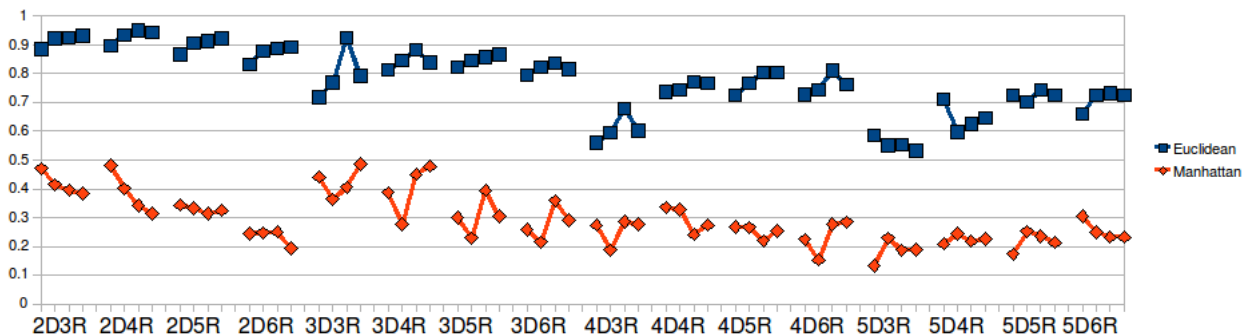


Figure 4: Compare distance metrics: Euclidean versus Manhattan

This graph plots the same set of experiments shown in Figure 3, but instead of bars, we plot the quality measures for each cluster as lines. Each small line segment corresponds to a cluster in the Figure 3, e.g. $5D3R$ is the same as the $3Ref$ cluster of $5D$ graph. The upper lines in this figure are the performances of Euclidean distance, which is the same as the values in Figure 3. The lower line segments are the results when we employ the Manhattan distance instead.

different data points in the original space will overlap in the transformed space.

At the current stage, we have yet to work out a formal derivation of optimal positions for the reference points that would minimise the loss of information after data transformation. Although we do not have a formal proof, we do have 2 general rules that can yield a good performance in practise:

1. Do not place the points too close to each other, because this will result in data points having similar distances to each reference point. An extreme case would be to place all reference points at the same spot, which would be useless. Therefore, one should place the reference points far away from each other, preferably outside the dataset region.
2. The layout of the reference points should aim to produce a unique set of distances for different data points. This can help to reduce the information loss, because, at least, different data points in the original space will not overlap in the kR space.

Based on these two rules, we placed our reference points around the maximum and minimum attribute values, and we also need to induce some randomness to ensure that the reference points do not have the same attribute values. In addition, we followed certain shape layout to make sure that the reference points are far away from each other, e.g. using a triangular layout for 3 reference points. An example set of 3 reference points to transform a 3D dataset would be $\{(0.10, 0.07, 0.98), (0.09, 0.95, 0.05), (0.97, 0.03, 0.04)\}$ (assuming the attributes are real numbers between 0 and 1).

3.2.4 Factor 2: Number of reference points

Figure 2 shows that it requires at least 3 reference points to uniquely identify data points in a 2D space, but will the same observation be made also in a higher dimensional space? Unfortunately, this is another difficult theoretical question that we do not yet have an answer to, but in our experiment it seems to suggest that 3 reference points can still produce a unique set of distances in a higher dimensionality to differentiate different data points. We have found that, within each test dataset, every set of distances is unique. So, 3 reference points can still uniquely identify every possible data points in a 5D space.

Although unique identification was possible, the number of reference points used still seems to have an impact on the quality of the transformed space. As shown in Figure 3, using $3Ref$ seems to be more unstable and worse than other choices of reference points. This effect is more so in higher dimensional spaces.

Another observation from Figure 3 is that, as the number of reference points increases, the quality of the dataset does not seem to get better, e.g. $4Ref$, $5Ref$ and $6Ref$ all seem to have comparable qualities regardless of their dataset dimensionality. We suspect this is due to the layout of the reference points, because although we are introducing more reference points, they are probably not at their optimal positions, so they may have been under-utilised.

3.2.5 Factor 3: Original Dimensions

This factor seems to be the most dominating, as suggested by the results in Figure 3. When the dimensionality increases, the quality of the transformed space generally decreases.

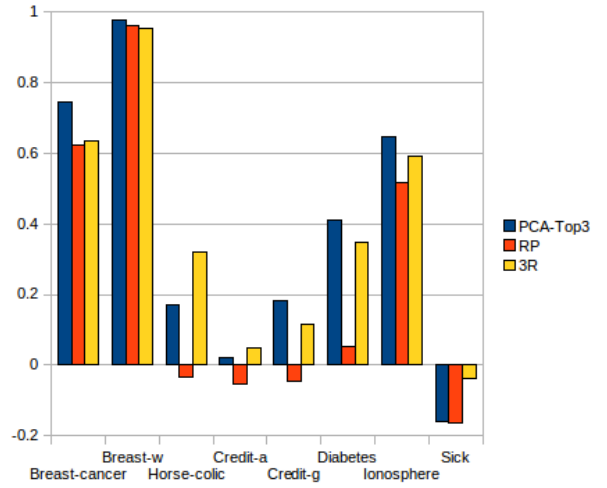


Figure 5: Comparison of transformation quality. The y -axis is the quality measurement described in Section 3.2. We compare PCA-Top3 (the first 3 principal components), random projection (RP) and 3 reference points ($3Ref$) in their ability to preserve the proximity of the data points.

3.2.6 Factor 4: Distance metrics

We tested two different distance metrics: Euclidean distance and Manhattan distance. For this experiment, in the quality measurement process, we used the same distance metric as the one used for the dataset transformation, i.e. the Euclidean distance will be evaluated with Euclidean distance in the quality measure, and the same goes for the Manhattan distance.

We used the same synthetic test datasets to compare the two distance metrics. The results are presented in Figure 4. We have found that the Euclidean distance is consistently better than the Manhattan distance in all settings.

4 Visualisations

For real world dataset visualisations, we used binary class problems from the UCI data repository (Asuncion and Newman, 2007). Some basic information about the datasets is listed in Table 1. In our experiments, we have used three tasks to compare our approach with PCA and RP. The first task compared their ability to preserve the proximity of the data points with the quality measurement. The second task compared the changes in the decision tree's learning performance, before and after the data transformation; the learning performance is measured with the area under ROC curve (Fawcett (2004)). In the third task, we looked at the actual visualisations they produced and see whether insights on the datasets can be gained.

Although 2D graphs look better on paper, but the minimum requirement for $kRef$ to work is with 3 reference points, therefore, for all tasks, we will reduce the dimensionality to 3 for visualisation. The visualisations are done in an interactive environment¹, but, unfortunately, this is difficult to present on paper. We will try to minimise the loss of interaction by rotating the 3D models to the most clear angle that we can find. However, the depth information will be lost.

¹We used mayavi for building the visualisations (<http://mayavi.sourceforge.net/>).

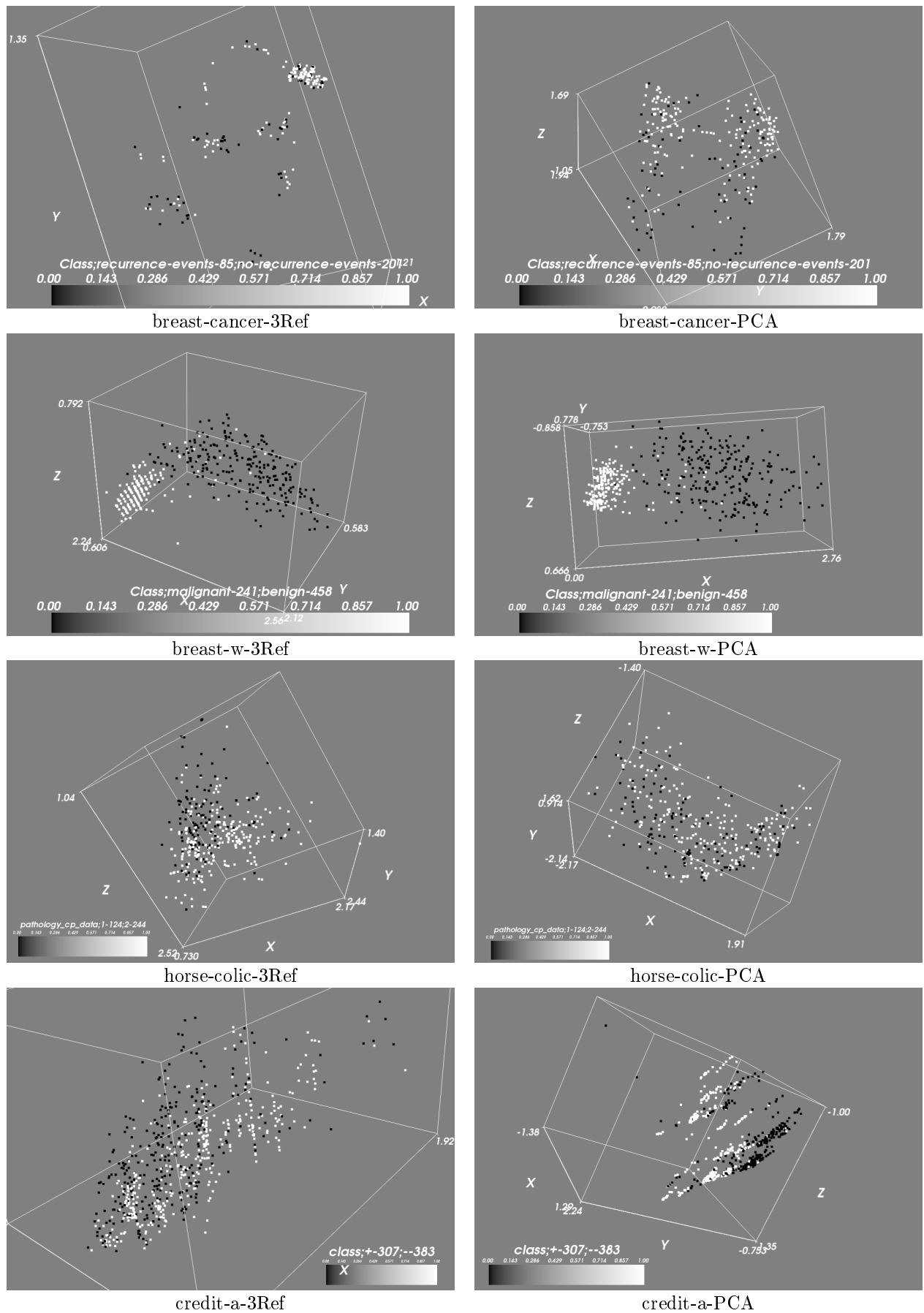


Figure 6: Comparing visualisations (Part1)
Comparing visualisations produced by 3 reference points (3Ref) and by PCA on 2-class problems from the UCI datasets.

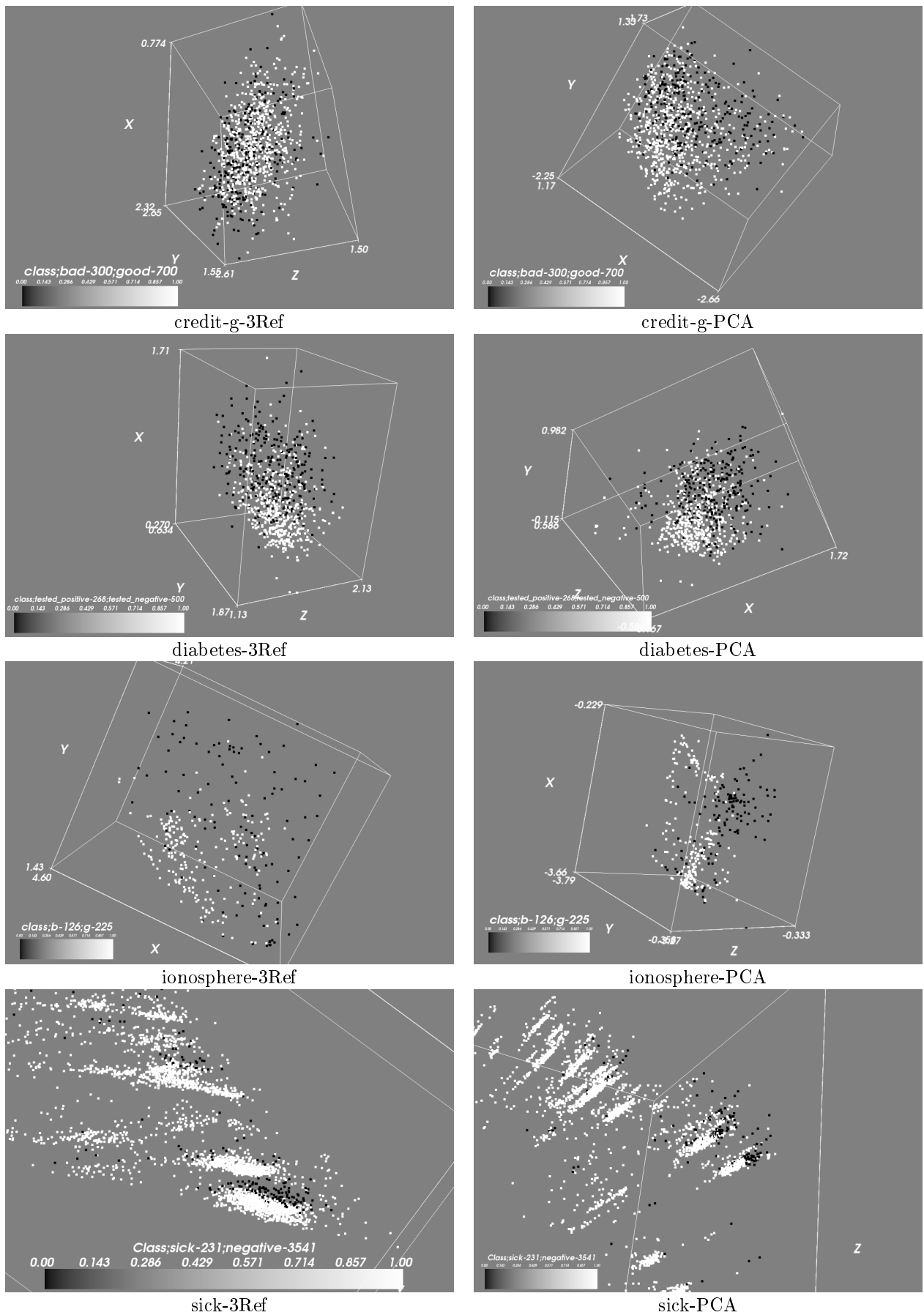


Figure 7: Comparing visualisations (Part 2)
Comparing visualisations produced by 3 reference points (3Ref) and by PCA on 2-class problems from the UCI datasets.

datasets	AUC	Numeric	Nominal	Positive%	Size
Breast-cancer	0.58	0	9	29.72	286
Breast-w	0.96	9	0	34.48	699
Horse-colic	0.49	7	20	33.70	368
Credit-a	0.89	6	9	44.49	690
Credit-g	0.64	7	13	30.00	1000
Diabetes	0.75	8	0	34.90	768
Ionosphere	0.89	0	34	35.90	351
Sick	0.95	7	22	6.12	3772

Table 1: Dataset information

This table shows some basic information about the datasets, including the performance of J48 (WEKA's implementation of C4.5 (Witten and Frank (2005))) in the area under the ROC curve (AUC). *Numeric* and *Nominal* denote the number of numeric attributes and nominal attributes in the dataset. *Positive%* is the percentage of the smaller class. *Size* is the total number of examples in the dataset.

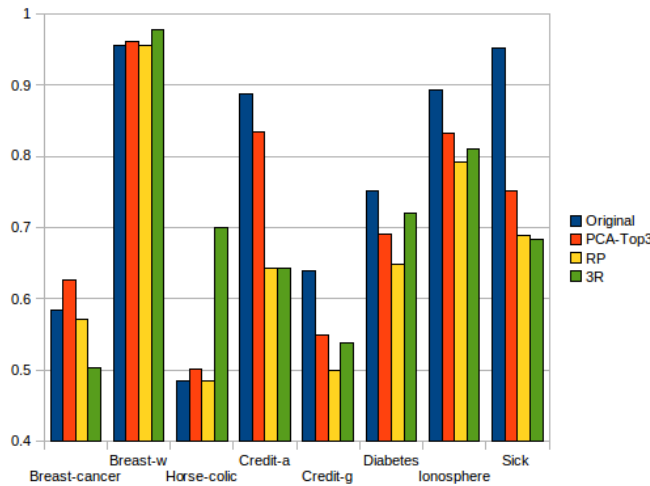


Figure 8: Compare the AUCs of J48

This figure shows how different data transformation impacts the area under the ROC curve. There are 8 bar clusters, representing 8 datasets. In each cluster, there are 4 bars, from left to right, denoting the AUC measures of the original dataset, PCA-Top3, RP and *3Ref*.

4.1 Task 1: Comparing quality measurements

The result for the first task is shown in Figure 5. The result shows that PCA is better in most cases except for *horse-colic*, *credit-a* and *sick*, but *3Ref* is not much behind. On the other hand, RP is the worse of the three in 7 out of 8 cases.

Since the quality measurement is the correlation between the before and after transformation distances, therefore a negative correlation means that the closer distances are getting farther, whereas the farther distances are getting closer. This scenario is considered poor quality, because it distorts the original image of the dataset. Negative numbers occurred 4 times in RP, this indicates that RP is distorting the dataset more than the other two methods. In the sick dataset, all three methods are negative, suggesting this dataset is quite difficult to preserve. The cause of difficulty is the abundance of binary attributes in the sick datasets, because in the distance calculation, binary attributes offers less numerical variations than numeric attributes. Therefore, it is harder to retain an accurate pairwise distances.

4.2 Task 2: Comparing learning performances

The results of the second task is presented in Figure 8, which shows that in 5 out of 8 cases, all dimensionality reduction techniques cause the learning performance to go down. If we compare *Original* with the next best AUC value, there are 2 cases, *credit-g* and *sick*, where the drop in AUC is relatively bigger than other cases. The drops in AUC are 0.09 for *credit-g* and 0.20 for *sick*. In both cases, their quality measurements in Figure 5 are also lower than others, except *credit-a*. The biggest AUC drop of 0.20 from sick dataset coincides with the result from task 1: the sick dataset does not project well onto lower dimensions.

Although the performances were dropped, but in most cases the drops were moderate, and because decision tree learner is learning based on the geometric information of the classes, therefore, it suggests that the data transformation technique is able to retain most of the geometric information, i.e. the relative geometric positions.

If we compare *3Ref* with others, *3Ref* does worse than PCA in 5 out of 8 cases, but does better than RP in 5 out of 8 cases with 1 tie.

4.3 Task 3: Comparing images

For the third task, we still used the AUC of a decision tree learner as a guide to help compare how much class information is preserved in the visualisations. This task is similar to task 2, but in this task, we try to stress that there are useful information when one has access to the actual visualisation. Task 2 was a quantitative analysis, but a lot of information about the geometrical structure was lost. In this task, without defining a specific aspect of the dataset to quantify, we used the dimensionality reduction techniques as a general purpose dataset visualiser.

In Figure 6 and 7, we present images of the UCI datasets with *3Ref*'s images on the left-hand side and PCA's images on the right-hand side. The colour of the classes is consistent in each pair of images, i.e. in both *breast-cancer-3Ref* and *breast-cancer-PCA*, the black data points refer to the same class. In addition to the images, Table 1 provides the performance measurement of the decision tree on the same set of dataset. The observations for each pair of images are as follows:

Breast-cancer We believe that *3Ref* has the better image, because it shows different clusters for the black class (recurrence-event class); the white class also seems to form different clusters. The dataset looks more clear cut in the *3Ref* image than in the PCA image. We can also isolate the difficult region, which seems to be the big cluster at the top right corner in *3Ref*'s image. In PCA's

image, the dataset seems to be divided into two symmetrical clusters.

Breast-w Both images separate the black and the white classes into one sparse cluster and one dense cluster. The dataset looks reasonably easy to learn and this is confirmed by the good learning performance.

Horse-colic Both images seem to be more complex, because the black class and the white class are mixed together. This difficulty is also confirmed by the decision tree's performance. *3Ref*'s image suggests that the difficulty comes from the centre region where classes are more mixed, whereas in the PCA's image, the points are evenly mixed with no obvious clusters.

Credit-a The image of PCA seems to be better than *3Ref*, because PCA shows more separation between the two classes and the decision tree is also able to learn quite well from the dataset.

Credit-g This is another difficult dataset. Both images are complex and the classes are overlapped. Both images seem to show two clusters, and the source of difficulty lies within the middle region, where the two classes overlap.

Diabetes The images are still complex, but both images seem to show that the white class is concentrated more at a specific region. So it looks like the errors are coming from the invading white points into the sparse black cluster.

Ionosphere This dataset seems to be sparse in *3Ref*'s image. The white class forms two well-separated clusters, while the black class is sparsely distributed. It is not difficult to learn, because the classes are not mixed. Although it can be hard to see, but the two classes actually lie at a different depth in *3Ref*'s image. PCA seems to be clustering the classes more tightly, and it also shows two white clusters.

Sick This imbalanced dataset has a very skewed class distribution (6% rare class), but the learning performance is quite good. The image from *3Ref* shows that the black class (rare class) forms a well-separated cluster, sitting just above a big white cluster. Similar pattern can be seen in PCA's image, but the division between the classes is not as clear as in *3Ref*'s image.

We did not show RP's images because RP's images were not as informative since the data points are tightly packed into lines or planes. For example, when we applied RP on the breast-cancer dataset, as shown in Figure 9, RP produced 4 lines of points, which is comparatively less rich in geometric variation than the other two methods.

5 Discussions

In our experiments, we have compared the correlation scores and the change in learning performance of *kRef* with PCA and RP. We have demonstrated that *kRef* transformation is capable of producing comparable visualisations of datasets. We have also shown that data visualisation can be a useful tool for understanding datasets, and it can also be more informative than a single learning performance measurement.

When producing data visualisations, there exists a tradeoff between the accuracy and the computational speed, although *kRef* is slightly less accurate than PCA, *kRef* is much faster. With a little sacrifice in accuracy, *kRef* is a better choice in practise because large datasets are abundant.

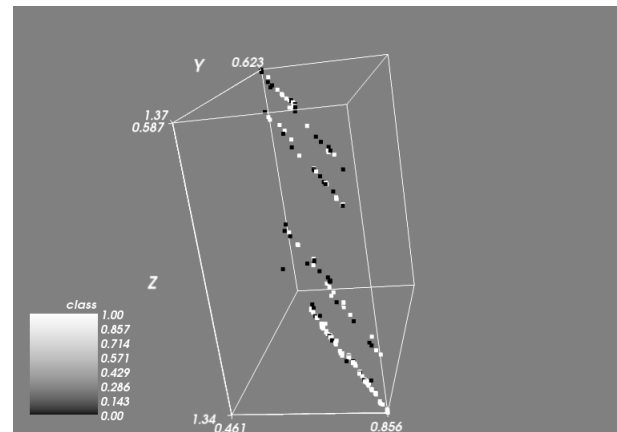
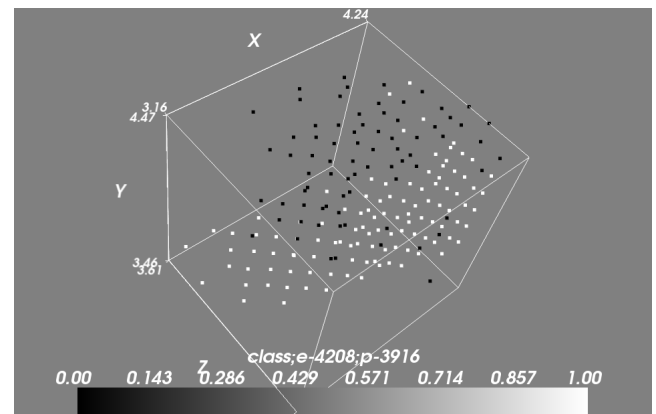
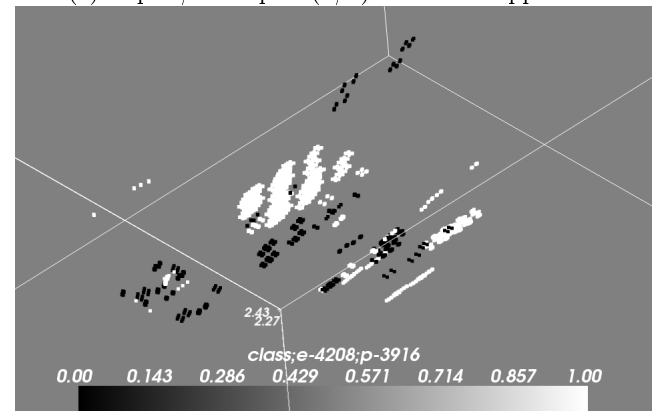


Figure 9: Breast-cancer dataset with RP
Perform random projection on breast-cancer dataset.
The data points form 4 lines in the RP space.



(a) "equal/not-equal (0/1)" distance approach



(b) VDM distance approach

Figure 10: Dealing with nominal attributes
These two figures are used to show the difference between the two different approaches of handling nominal attributes. The dataset used is the mushroom dataset from the UCI data repository. Black points represent the *edible* class and white points represent the *poisonous* class. The dataset size is 8124, but in figure (a), the 0/1 approach has placed a lot of data points at the same spot, making it rather difficult to see all 8124 different data points. On the other hand, the VDM approach in figure (b) is able to spread out the data points more so we can see more interesting trends in the datasets.

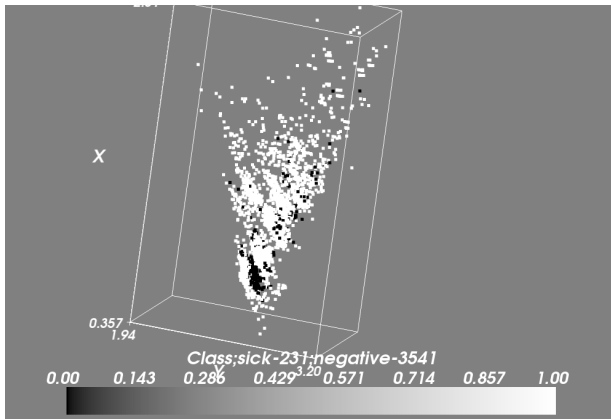


Figure 11: The data points distribution shape The *sick* dataset looking from afar. The dataset shows a cone shape distribution in the *kRef* space.

5.1 Implementation issues

The first issue was how to deal with nominal attributes, because they may not be ordinal attributes. In our first attempt, we used the simple equal/not-equal approach: assign 0 distance if the two attributes have the same value, and assign 1 otherwise. Unfortunately, this approach produces very dense clusters, therefore making it hard to visualise because a lot of points are packed at the same point. So, we tried Value Difference Metric (VDM) distance (Wilson and Martinez, 1997), which essentially replaces nominal values with their respective occurrence rates. The VDM distance approach is able to give data points more variations, which allows the data points to spread out, thus leading to better visualisations. The mushroom dataset is used as an example to show the difference between the two approaches. Mushroom dataset has 22 nominal attributes and 8124 examples. It is considered to be one of the easier datasets in UCI, which is also evident from our visualisation. Figure 10(b) shows that the two classes are well separated and both classes have clean clusters that do not overlap. It should be noted that the black and white groups in Figure 10(b) are sitting at a different depth, and only appears to be overlapping because of the 2D perspective.

The second implementation issue was that: if there are numeric attributes that have large values, then the distance measurement will be dominated by these numeric attributes. So, in order to allow an even contribution from all attributes, we need to normalise the numeric attributes.

5.2 *kRef* space restriction

There are regions in the transformed *kRef* space that will not be used, for example, under a triangular layout, it would be impossible to have a data point that has 0 distances to all three reference points. So, this suggests that the transformed space will have a hyper-concave shape distribution, where the bottom is sitting at the centre of the reference points. As shown in Figure 11, a zoomed out image of the *sick* dataset, the transformed dataset is distributed in a cone shape.

5.3 Comparison with other methods

kRef and PCA share similar concepts; they both rely on data point projection, but the difference is that: in *kRef*, the projection is done via different points; whereas in PCA, the projection is done via different eigenvectors (principal components). In the

kRef method, each data point in the datasets is re-described by all the reference points, but in PCA, the eigenvectors can be used independently. The success of the *kRef* method relies on reference points to generate distance with the highest variance so that the data points will not overlap in a condensed space. Similarly, PCA tries to project onto eigenvectors that would reflect the highest variance.

In terms of computation, *kRef* has a better run-time than PCA. The run-time of the *kRef* method is $O(mn)$, given m number of examples and n number of features. In comparison, PCA is more expensive to compute, with a run-time of $O(\min(mn^2, nm^2))$. Thus, given a large enough m and n , the run-time of *kRef* method is $O(m^2)$, whereas PCA runs at $O(m^3)$. The run-time of PCA is derived from the run-time of Singular Value Decomposition (SVD), because computing the covariance matrix directly would be too costly if the dataset dimensionality is high, e.g. 10000 features would require a matrix size of 10000x10000. So, the run-time of PCA is dominated by SVD computation, which has a run-time complexity of $O(\min(mn^2, nm^2))$.

Another advantage of the *kRef* method over PCA is that the *kRef* method can run in parallel, because most of the computations in *kRef* are not dependent, so the calculations can be distribute across different machines. This parallel distribution is quite attractive because large datasets can be partitioned and stored in separate machines, which is also more memory efficient.

The *kRef* method is similar to random projection as well, because they both transform the dataset in a similar fashion, but our approach is motivated by triangulation, so our random matrix selection is quite different from the ones used in random projection. Also, our method uses distance function in the projection calculation, whereas random projection uses dot products.

We have implemented random projection proposed by Achlioptas (2001) and tested it in our experiments. As shown in Figure 9, random projection generates less geometrical variances. This suggests that while random projection is useful as a dimensionality reduction method and enjoys a theoretical bound, its constraints on the random matrix makes it unable to provide good visualisations in 3D space.

5.4 Reflection

In order to address the question of visualisation meaningfulness, as proposed in the introduction, we must first define the *meaning* in the context of the visualisation. For example, in a stock OHLC (open,high,low,close) bar chart, the meaning could be the underlying supply and demand, which is what OHLC bar chart tries to visualise. In the context of this work, we define the meaning as the underlying geometric structure of the dataset. Therefore, for a given dataset, there exists a true “meaning” that different data visualisation methods all try to capture.

Our visualisation is meaningful because it is able to retain most of the geometric structure of the dataset. The accuracy of the *kRef* method is only slightly behind PCA, i.e. in 6 out of 8 cases in Figure 8 the gaps were within 0.1. Moreover, it is possible if a better layout of the reference points can be derived, the accuracy of the *kRef* method could be further improved.

Like any generic dimensionality reduction method, *kRef* will work with any number of dimensions or any dataset size, however, the usefulness of the visualisation depends on the users. Because visualisations are inherently subjective, the same visualisation may be

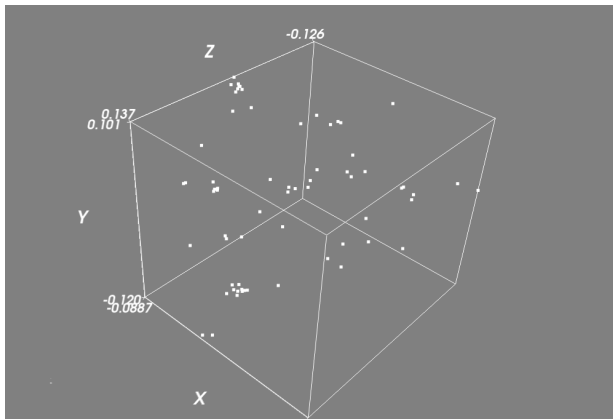


Figure 12: Visualise feature correlations for *audiology*. An illustration of the *kRef* method's extension work. In this visualisation, the features are treated as examples and the distance function used here is the correlation function. So, in this space, the clusters imply highly correlated features.

interpreted differently, which could lead to different insights. This creativity is the strength of the data visualisation methods.

5.5 Future works

The interactive aspect of the visualisation was not presented in this work because it would require a different evaluation process. However, an interactive environment may have features that give the users the ability to rotate the 3D model, zoom-in from any angle, select a sample of data points and check the data point information by clicking the dots. These interactions allow the user to explore and conceptualise the difficult parts of the dataset, which can potentially lead to a better formulated learning approach for the problem. So, it would be interesting to investigate this active learning approach and see how best to utilise the insights from the visualisations.

It is also possible to apply *kRef* to visualise the feature space instead of the example space. This can be done by taking a transpose of the dataset matrix, so the features become examples and examples become features. The similarity function also needs to be changed from the Euclidean distance function to a correlation function for a more meaningful interpretation of the relationship between features. Using the correlation as the similarity function, we can see correlated features as they form clusters in the *kRef* space. An example of visualising the feature space is presented in Figure 12, which shows the features of the *audiology* dataset from the UCI data repository. In this dataset, there are 69 attributes and the visualisation shows two obvious clusters, which suggests that there are two sets of correlated attributes in this dataset.

Other interesting future works include deeper investigation of theoretical issues about the reference point method, such as the optimal position for the reference points placement; or, applying this method as an efficient way of locating approximate nearest neighbours.

6 Conclusions

Visualisation is a useful tool for analysing difficult problems and in this work, we offered a new method to address the problem of visualising high-dimensional datasets. We compared our approach

with other generic methods, namely PCA and random projection, and the results suggest that PCA is a computationally expensive approach, while random projection delivers inaccurate visualisations. Our method, on the other hand, is able to produce accurate visualisations in a cost-effective manner. Therefore, we recommend this method as an attractive visualisation tool to assist the data mining process.

References

- Achlioptas, D. (2001), 'Database-friendly Random Projections', *Symposium on Principles of Database Systems*.
- Asuncion, A. and Newman, D. (2007), 'UCI machine learning repository', *University of California, Irvine, School of Information*.
- Bingham, E. and Mannila, H. (2001), 'Random projection in dimensionality reduction: applications to image and text data', *conference on Knowledge discovery and data mining*.
URL: <http://portal.acm.org/citation.cfm?id=502546>
- Blum, A. (2006), 'Random projection, margins, kernels, and feature-selection', *Lecture Notes in Computer Science* pp. 52–68.
- Borg, I. and Groenen, P. J. (1997), *Modern multidimensional scaling : theory and applications*, Springer series in statistics, Springer.
- Fawcett, T. (2004), ROC Graphs: Notes and Practical Considerations for Researchers, Technical report, HP Laboratories.
- Fodor, I. (2002), 'A survey of dimension reduction techniques', *Manuscript* pp. 1–18.
- Inselberg, A. (1996), Parallel Coordinates: A Guide for the Perplexed, in 'Hot Topics Proc. of IEEE Conference on Visualization', pp. 35–38.
- Johnson, W. and Lindenstrauss, J. (1984), 'Extensions of Lipschitz mappings into a Hilbert space', *Contemporary mathematics* **26**, 189–206.
- Jolliffe, I. T. (2002), *Principal Component Analysis 2nd Edition*, Springer Series in Statistics, Springer-Verlag.
- Kohonen, T. (2000), *Self-Organizing Map*, Springer.
- Lin, J. and Gunopulos, D. (2003), 'Dimensionality reduction by random projection and latent semantic indexing', *Proceedings of the Text Mining Workshop, at the 3rd*.
- Moore, D. (2006), *Basic Practice of Statistics*, WH Freeman Company.
- Morris, C. J., Ebert, D. S. and Rheingans, P. L. (2000), Experimental Analysis of the Effectiveness of Features in Chernoff Faces, in W. R. Oliver, ed., '28th AIPR Workshop: 3D Visualization for Data Exploration and Decision Making', Vol. 3905, SPIE, pp. 12–17.
- Wilson, D. R. and Martinez, T. R. (1997), 'Improved Heterogeneous Distance Functions', *Journal of Artificial Intelligence Research* **6**, 1–34.
- Witten, I. H. and Frank, E. (2005), *Data Mining: Practical Machine Learning tools and techniques, 2nd Edition*, Morgan Kaufmann, San Francisco.

FlowRecommender: A Workflow Recommendation Technique for Process Provenance

Ji Zhang¹, Qing Liu² and Kai Xu²

¹ Department of Mathematics and Computing,
University of Southern Queensland, Australia

Email: ji.zhang@usq.edu.au

²CSIRO ICT Centre, Hobart, TAS, Australia

Email: {[q.liu](mailto:q.liu@csiro.au), [kai.xu](mailto:kai.xu@csiro.au)}

Abstract

The increasingly complicated workflow systems necessitates the development of automated workflow recommendation techniques, which are able to not only speed up the workflow construction process, but also reduce the errors that are possibly made. The existing workflow recommendation systems are quite limited in that they cannot produce a correct recommendation of the next node if the upstream nodes/sub-paths that determine the occurrence of this node are not immediately connected with it. To solve this drawback, we propose in this paper a new workflow recommendation technique, called *FlowRecommender*. FlowRecommender features a more robust exploration capability to identify the upstream dependency patterns that are essential to the accuracy of workflow recommendation. These patterns are properly register offline to ensure a highly efficient online workflow recommendation. The experimental results confirm the promising effectiveness and efficiency of FlowRecommender.

1 Introduction

In recent years, workflow systems are becoming more and more complicated as a result of a fast growing number of scientific processes available. Scientific workflows are based on the automation of scientific processes in which scientific programs are associated, based on data and control dependencies [1]. These scientific processes, mostly taking the form of Web services, could either be local or remote scientific tools or programs that can be shared by scientists from a common domain. However, the construction of most workflows are based on some pre-determined templates and relevant domain knowledge plays a crucial role in creating these templates. As such, the workflow construction is difficult or even impossible when domain knowledge is missing or the workflows are to be constructed by amateurs in the field. Workflow recommendation based on provenance turns out to be a possible and promising approach in the case when no templates are available.

Provenance of workflows is a practice to archive historical workflows that have been executed and, sometimes, also the intermediate and final results generated by the workflow processes. A number of provenance systems and techniques have been proposed [6, 7, 8, 9, 10, 11, 12, 13, 14, 15]. The provenance of workflows is of considerable value to scientists. From it, one can ascertain the

quality of the data based on its ancestral data and derivations, track back sources of errors, allow automated re-enactment of derivations to update a data, and provide attribution of data sources [4]. Recent work has also shown that provenance information (the metadata required for reproducibility) can be used to simplify the process of pipeline creation [5]. Tools for assisting automatic construction of workflows are increasingly desirable to facilitate the construction of complicated workflows. An effective and efficient workflow recommendation technique is useful in these tools. First, it can speed up the workflow construction process by reducing the development time. Second, it can provide a guidance for choosing the mostly likely node and, therefore, minimize the errors that are possibly made in the workflow construction.

An important observation in workflow construction practice is that, in most cases, the prediction of a downstream node is only dependent on one of its adjacent upstream sub-paths in the workflow. Here, the adjacent sub-path is not necessarily continuous nor immediately connected the node. The influence exerted by the remote upstream sub-paths becomes negligible when the distance between the node and the upstream sub-path increases. The existing work perform recommendation either based on the last node only [1] or the continuous paths that ends in the last node in the current workflow [2]. They cannot perform recommendation based on the paths that are not continuous nor does not terminate in the last node of the current workflow. For example, suppose we need to produce the recommendation of the next node for a partial workflow $c \rightarrow a \rightarrow b$. The method in [1] provides prediction based on node b only while the method in [2] provides prediction based on one of the two continuous sub-paths that end at node b : $c \rightarrow a \rightarrow b$ and $a \rightarrow b$. These two methods will fail to provide correct recommendation if the next node is actually decided by other sub-paths such as c , a , $c \rightarrow a$ or $c \rightarrow \dots \rightarrow b$.

To solve the drawbacks of the existing work, we propose a new workflow recommendation technique based on workflow provenance, called *FlowRecommender*. FlowRecommender provides a more effective yet efficient means for producing the prediction by investigating the correlation of each possible workflow node with respect to its adjacent upstream paths. More specifically, FlowRecommender takes two main stages for performing workflow recommendation: the offline stage and the online stage. In the offline stage, FlowRecommender extracts the patterns of nodes that will appear in the workflows. The patterns are called the *influencing upstream sub-paths* of the nodes that determine the occurrence of these nodes in the workflows. The extracted patterns are registered into the so-called *pattern table* for the subsequent recommendation. In the online stage (when recommendation is required), the pattern table is scanned to match the patterns with the current partial workflow under construction. The node is recommended if its influencing upstream sub-path

Copyright ©2009, Australian Computer Society, Inc. This paper appeared at the Eighth Australasian Data Mining Conference (AusDM 2009), Melbourne, Australia. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 101, Paul J. Kennedy, Kok-Leong Ong and Peter Christen, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

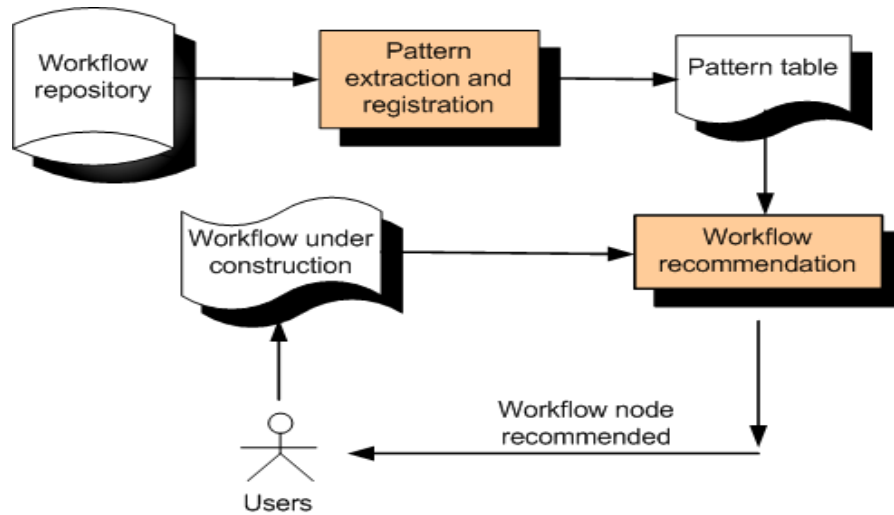


Figure 1: The system architecture of FlowRecommender

matches the partial workflow. Compared with the existing methods, FlowRecommender is advantageous in that it features a stronger capability to identify the influencing upstream sub-paths, leading to a better recommendation performance.

The reminder of this paper is organized as follows. Section 2 presents an overview of FlowRecommender and its system architecture. In Section 3, greater details are given on the two major modules of FlowRecommender, i.e. pattern extraction and registration, and the workflow recommendation. The experimental results are reported in Section 4. The final section concludes the whole paper and presents some future research directions.

2 An Overview of FlowRecommender

In this section, we will present an overview of FlowRecommender for workflow recommendation based on workflow provenance. The system architecture of FlowRecommender is presented in Figure 1. Generally, there are two modules involving in FlowRecommender, i.e., pattern extraction and registration and workflow recommendation. The first two modules are performed offline while the last module is conducted online during the construction of workflows.

- **Pattern extraction and registration.** The patterns of the candidate nodes are extracted from the workflows in the provenance. Here, the candidate nodes are those tools/programs that can be utilized to extend/complete the workflow in the recommendation, and the pattern for each candidate node refers to its influencing upstream sub-path that determines the occurrence of this node. Such pattern is identified when the correlation (measured by *confidence*) between the sub-path and the node is sufficiently strong. The discovered patterns are registered into the *pattern table*, making FlowRecommender ready for the subsequent workflow recommendation module;
- **Workflow recommendation.** During workflow construction, workflow recommendation module tries to match the influencing upstream sub-paths of the candidate nodes against the current workflow under construction. The nodes are recommended to users once its influencing upstream sub-path matches the current workflow.

From the system architecture of FlowRecommender as shown in Figure 1, we can see that, in the workflow construction process, a close cycle is formed among the fol-

lowing components: the workflow currently under construction, the workflow recommendation generation module and the end human users. The workflow currently under construction serves the input to the recommendation generation module. Based on the current status of the workflow, the recommendation generation module tries to provide recommendation as to which node should be selected to extend/complete the workflow. The recommendation results are fed to the users who will decide whether the recommendation is followed. The users' decision will lead to the extension of the workflow. This cycle continues until the workflow has been constructed to the point such that the desired task has been fulfilled.

The major constituting modules of FlowRecommender are discussed in details in the following subsections.

2.1 Pattern Extraction and Registration

In this section, we will discuss in details how to extract patterns from provenance that are useful to the workflow recommendation. These patterns serves as a sort of signatures to activate the recommendation of certain nodes to extend/complete the partial workflows.

2.1.1 Pattern Extraction

The patterns of the candidate nodes are extracted from the provenance. The candidate nodes are those nodes that can be potentially used to extend/complete the partial workflows under construction. The patterns are the influencing upstream sub-paths that determine the occurrence of nodes in the workflows.

Definition 2.1: Candidate Node Set for all the workflows. The Candidate Node Set with respect to all the workflows in the provenance, denoted as $CNS(D)$, is the set of nodes that can be potentially recommended in various locations of workflows. It is defined as the set of nodes that have appeared in the workflows but do not only appear in the start position of the workflows.

Definition 2.2: Upstream sub-paths. The upstream sub-paths of a node v in a workflow w is defined as the sequences of ordered nodes that appear before v in w . For example, in a workflow $b \rightarrow a \rightarrow c$, the set of upstream sub-paths for node c are $\{b, a, b \rightarrow a\}$.

We evaluate the correlation of a node and its upstream sub-paths through the measure of *confidence*. Confidence of a node v given a upstream sub-path p is the probability that v appears given that p has already appeared in the

workflow. It is defined as

$$Conf(v, p) = \frac{freq(v, p)}{freq(p)}$$

where $freq(v, p)$ and $freq(p)$ correspond to the frequency/count that v and p occur together and p occurs alone in the workflow, respectively.

Unlike association rules, only confidence is leveraged in our work to measure the significance of the patterns extracted, instead of using both support and confidence. In workflow domain, it is likely that some workflows are executed in a quite low frequency, but their constituting nodes and/or paths feature strong correlations with other. If support measure is used, then it may lead to many low-frequency workflows being screened out and the recommendations based on these workflows becomes impossible.

Definition 2.3: Influencing upstream sub-path. For a node v , the influencing upstream sub-path p in a workflow is defined as the sub-path that satisfies that the confidence of v given p is no less than a given confidence threshold σ_{conf} , as follows:

$$Conf(v, p) \geq \sigma_{conf}$$

The technical challenge lies in extracting the influencing upstream sub-path for a node v is that we are not able to accurately pinpoint the location and order of the influencing upstream sub-path of v . The concept of the location of an influencing upstream sub-path p in a workflow is relative to the end (last node) of this workflow. This is what we call the backward location of a sub-path within a workflow, which is defined as follows:

Definition 2.4: Backward location of an influencing upstream sub-path. The backward location of an influencing upstream sub-path p within a workflow is defined as the distance (i.e., the number of edges) between the first node of p and the last node of w , i.e.

$$Location(p) = Dist(start(p), end(w)), p \in w$$

To solve the difficulty in pinpointing the location and order of the influencing upstream path of given nodes, we devise a technique to do this in a *progressive* fashion. For a given node, the technique first evaluates its confidence with respect to its upstream sub-paths consisting of nodes with the smallest overall distance (based on the location of the sub-path within the workflow as defined in Definition 2.4) to ensure that the more adjacent upstream paths are evaluated first, followed by the more remote ones. In other words, we evaluate the sub-paths with a distance of 1, 2, ... For the sub-paths with the same distance from the end of the workflows, we will first evaluate those with a smaller order. In the case of a tie of the order of constituting nodes in the paths, the algorithm will randomly choose a path for evaluation. This design is consistent with the rationale that the influencing upstream path of a candidate node is relatively close to the location of the candidate node in the workflow.

As the algorithm will potentially evaluate all the possible upstream sub-paths, thus the total number of such sub-paths could be large especially when the sub-path is far from the node in the workflow. To prevent an explosion of the possible upstream paths, a parameter k ($k \geq 1$) will be used to specify the maximum backward location for the sub-paths to be evaluated. In other words, k will determine the extent to which the upstream back-track will be performed to find the influencing upstream sub-paths for the given candidate node. For a candidate node, the upstream sub-path exploration is continued until either of the following conditions is met:

1. The influencing upstream sub-path of this node is found; or

2. All the sub-paths with a backward location not exceeding k have been evaluated.

The algorithm for finding the influencing upstream sub-paths for all the candidate nodes is presented in Figure 2. To speed up the pattern extraction, particularly the calculation of confidence, we leverage *inverted indexing* of workflows, which speeds up the search of the workflows where a given node appears. This can significantly reduce the number of workflows to be evaluated for calculating confidence. The inverted indexing based on the candidate nodes are first performed in order to streamline the subsequent confidence calculation. $CNS(D)$ is cloned to $SetOfPendingNodes$ which will dynamically updated in the algorithm to track the set of nodes whose patterns have not yet been found. The FOR loop in Line 3 controls the order of the sub-paths the algorithm will evaluate, increasing from 1 through k . The algorithm will continue when not all the candidate nodes have been evaluated. Once the influencing upstream sub-path for the node has been identified, the node and its pattern will be registered into the pattern table and the algorithm will start to process the next candidate node. Pattern registration in pattern table will be discussed in the next subsection. The BREAK clauses in Line 10 enables the algorithm to be terminated early the moment when the influencing upstream path has been identified with respect to each candidate node.

2.1.2 Pattern Registration

When they have been extracted, the influencing upstream sub-paths of the nodes in Candidate Node Set for the provenance will be registered into the pattern table. Next, we will present the definition of pattern table.

Definition 2.4. Pattern Table. The pattern table is an $n \times 2$ table, where x_{i1} is the node that possibly appears in the workflows (i.e., $x_{i1} \in CNS(D)$) and x_{i2} is the corresponding influencing upstream sub-path of the node given in x_{i1} , where $1 \leq i \leq n$.

An influencing upstream sub-path is represented as an ordered sequence of nodes in the pattern table. Each node in the sequence is associated with the location information represented by its distance to the candidate node given in the field of x_{i1} . The order of this sequence of nodes is consistent with the order they appear in the workflows from where they are extracted, but they do not necessarily appear consecutively. The pattern table is pre-constructed before recommendation is performed.

An example of pattern table is given in Table 1. Suppose that this table is derived from a repository of workflows involving a total of 7 nodes (labeled as a, b, c, d, e, f and g) and there are, however, only 3 nodes (i.e., c, d and e) whose influencing upstream sub-paths are identified given a certain confidence threshold level: the 2-order sub-path $a(3) \rightarrow b(1)$ is identified for node c and 1-order sub-path $c(2)$ and $g(1)$ are found for nodes d and e , respectively. The influencing upstream sub-path of node c , i.e., $a(3) \rightarrow b(1)$, means that node c is recommended when the current workflow under construction contains a path that takes the form of $* \rightarrow a \rightarrow ? \rightarrow b$, where the wildcard symbol asterisk(*) represents a sub-path with any possible sequence of nodes while the question-mark(?) represents a single node.

2.2 Workflow Recommendation

The workflow recommendation that our method provides is offered in a stepwise fashion; the systems automatically recommends the next most likely node to choose in order to extend/complete the current workflow that is under construction. The users can exert to activate/inactivate workflow recommendation anytime in the construction process, providing users with a great flexibility to choose construction with or without automatic workflow recommendation.

Algorithm: findInfluencingSubPath(D, k)**Input:** The whole workflow repository D and the limit k for upstream back-tracking for identifying patterns.**Output:** The influencing upstream sub-paths of nodes in $CNS(D)$.

1. Perform inverted indexing based on the nodes in $CNS(D)$;
2. $SetOfPendingNodes \leftarrow CNS(D)$;
3. FOR $i = 0$ to $k - 1$ DO
4. FOR each node v in $SetOfPendingNodes$ DO
5. FOR each sub-path p of backward location of i in the workflows w of $index(v)$, starting from the smallest order, DO {
6. $Conf(v, p) \leftarrow computeConfidence(v, p, D)$;
7. IF $Conf(v, p) \geq \sigma$ THEN {
8. Register p for v in the pattern table;
9. Remove v from $SetOfPendingNodes$;
10. BREAK; }

Figure 2: The algorithm for finding the influencing upstream sub-paths for candidate nodes

Candidate node label	Influencing Upstream Sub-path
c	a(3)→ b(1)
d	c(2)
e	g(1)

Table 1: A sample pattern table

Definition 2.5: Candidate Node Set for a workflow.

The Candidate Node Set for a workflow w , denoted as $CNS(w)$, is the set of nodes that can be potentially recommended to extend/complete an incomplete workflow w that has been constructed. It is defined as the set of nodes that satisfy the I/O constraints w.r.t w . That is, the input data type of the node in $CNS(w)$ matches the output data type of the last node of w . Obviously, we have $CNS(w) \subseteq CNS(D)$, and $CNS(w)$ may change when w is constructed at different stages.

The moment when the recommendation is required to extend or complete a workflow w , we need to go through evaluating the influencing upstream sub-path of each node in $CNS(w)$, which have been stored in the pattern table, to see whether they match the current workflow under construction. To perform pattern matching, we need to first define the distance between a partial workflow w and an influencing upstream sub-path p of a candidate node. Specifically, such distance, denoted as $Dist(w, p)$, is defined as the normalized sum of the location difference between the same pair of nodes in w and p as

$$Dist(w, p) = \frac{\sum Dist(n_i^w, n_j^p)}{Order(p) \cdot Order(w)}, n_i^w \in w, n_j^p \in p$$

where n_i^w and n_j^p represent the same node in w and p with (probably) different locations within w and p , $1 \leq i \leq |w|$ and $1 \leq j \leq |p|$.

Based on the above definition, we know that $0 \leq Dist(w, p) < 1$. We have $Dist(w, p) = +\infty$ if w does not have the same sequence of nodes appearing in p for a candidate node. This is to ensure that the partial workflow w and its matched influencing upstream sub-path p have the same sequence of nodes, though these nodes may have (slightly) different locations within the workflow and sub-path.

A distance threshold, denoted as σ_d , needs to be specified to determine whether the partial workflow w matches the influencing upstream sub-path p of a candidate node. That is, if $Dist(w, p) \leq \sigma_d$ then we say that w matches p and does not otherwise. σ_d is a parameter providing flexibility for controlling the accuracy/fuzziness in pattern matching. The larger σ_d is, the less accurate (more fuzzy) the matching will be, and vice versa.

If the patterns are matched for more than one candidate downstream nodes, then the recommendation can be presented in a *probabilistic* manner. Specifically, suppose $matchedCNS(w)$ is the set of matched candidate nodes

that satisfies that

$$matchedCNS(w) \subseteq CNS(w)$$

and

$$\forall v \in matchedCNS(w), Dist(w, p) \leq \sigma_d$$

where p is the influencing upstream sub-path of node v . Each node in $matchedCNS(w)$ will be recommended with a probability to indicate the strength that this node is recommended. The probability is quantified proportionally based on the confidence level, i.e.,

$$Strength(v, w) = \frac{Conf(v, w)}{\sum_i Conf(v_i, w)}$$

where $v_i \in matchedCNS(w)$. $strength(v_i, w)$ satisfies that $0 < strength(v_i, w) \leq 1$ and $\sum_i strength(v_i, w) = 1$.

If no influencing upstream sub-path can be matched against the partial workflow under construction for any candidate node, then only the nodes that satisfy the I/O interface of the workflow will be recommended (i.e., the output datatype of the last node of the workflow matches the input datatype of the node to be recommended), each with the same strength of $\frac{1}{|CNS(w)|}$, where $|CNS(w)|$ denotes the number of nodes in $CNS(w)$.

The algorithm for the recommendation generation is presented in Figure 3. Two sets, $SetOfRecommendedNodes$ and $SetOfStrength$, are used to record the set of nodes whose patterns matched the workflow and their respective recommendation strength, respectively. These two sets are initialized as empty sets at the beginning (Step 1 and 2). The pattern table is then scanned to identify those nodes whose patterns match the partial workflow and these nodes are stored in $SetOfRecommendedNodes$ (Step 3-5). Their strength in the recommendation is calculated and stored in $SetOfStrength$ based on their confidence level. Finally, the recommendation is presented by returning $SetOfRecommendedNodes$ and $SetOfStrength$ to users.

3 Experimental Evaluation

In this section, we present experimental evaluation of the our workflow recommendation technique. Three major

Algorithm: RecommendationGeneration(w)**Input:** A partial workflow w .**Output:** The set of nodes recommended for w and their respective strength.

1. $SetOfRecommendedNodes \leftarrow \emptyset$;
2. $SetOfStrength \leftarrow \emptyset$;
3. FOR each node v registered in the pattern table DO
4. IF $Dist(w, p) \leq \sigma_d$, where p is the pattern of v , THEN
5. $SetOfRecommendedNodes \leftarrow \cup v$;
6. FOR each node $v \in SetOfRecommendedNodes$ DO
7. $SetOfStrength \leftarrow \cup \frac{Conf(v, w)}{\sum_i Conf(v_i, w)}$, where $v_i \in SetOfRecommendedNodes$;
8. Output $SetOfRecommendedNodes$ and $SetOfStrength$;

Figure 3: The algorithm for producing workflow recommendation

sets of experiments are carried out, evaluating the accuracy of recommendation, scalability towards large provenance, and sensitivity to the major parameters. The program is developed in C++ and all the experiments are conducted in Windows Vista 2.26GHz system with a main memory of 2G.

The workflow provenance that we will use in the experiments are generated synthetically. To render the workflow repository generated as being close to the real-life application scenarios as possible, four major aspects of design are carefully considered in the designing process: a) What is the total number of workflows in the provenance (denoted as $N_{provenance}$)? b) What are the nodes that will appear in the workflow provenance (here, the total number of nodes that will appear in the provenance is denoted as N_{nodes})? c) What is the length of each workflow? and d) What is the order of nodes appearing in each workflow?

Both $N_{provenance}$ and N_{node} can be easily specified as positive integers. Once N_{node} is specified, the generator automatically generates a set of nodes as $P_1, P_2, \dots, P_{N_{node}}$. A maximum length of workflows, denoted as l_{max} , is specified and the length of each workflow is a random integer variable in the range of $[1, l_{max}]$.

To decide the order of nodes appearing in workflows, a set of matrices are constructed to decide the transitional probability for each pair of nodes. Specifically, each entry $x_{a,b}$ in the matrix $\mathcal{M}(i)$ corresponds to the transitional probability of node b given node a that is of a distance of i before b . Here, i is an integer and, without losing generality, we set it in the range of $[1, 3]$, meaning that the occurrence of a particular node in the workflows we generate is depended on a preceding node that is of a distance not exceeding than 3. Certainly, one may specify i as another valid value. Each workflow is initialized using a node randomly chosen from the set of nodes. When each subsequent node needs to be generated in the workflow, a matrix is randomly chosen from the matrix set (which contains three matrices) and the new node is generated based on the transitional probability presented in this matrix. This design ensures that occurrence of a node within workflows is not only determined only by its immediately preceding node, but also some more remote non-connected nodes. The workflow grows in this way until its specified length is reached.

3.1 Effectiveness Study

In order to carry out effectiveness study, we need to have a mechanism to validate the accuracy of the recommendation provided by FlowRecommender. To this end, we sample a small fraction of the workflows from the provenance (e.g., 10%) to evaluate the effectiveness of recommendation of FlowRecommender. This set of workflows is called the *test set*. For each workflow in this test set, a so-called *test node* is randomly chosen. The test node, which is the node that has really been executed, will be compared with the recommendation produced using FlowRecommender.

The effectiveness of recommendation is measured by

the accuracy of recommendation. Because the nodes in the workflows are generated stochastically based on the transition probability matrices, thus we will consider the top m recommended nodes when we evaluate the recommendation accuracy. A hit (i.e., accurate recommendation) is counted as long as the test node is one of the top m recommended nodes by FlowRecommender. We compare recommendation accuracy of FlowRecommender with that of other two competitive recommendation methods. The first method recommends the next likely node based on its immediately preceding single node, and the other one performs recommendation based upon the immediately preceding continuous upstream sub-paths. The comparison result is shown in 4. The value of m is set as 3 in this experiment. This figure shows that FlowRecommender performs much better than the method recommends node based on immediately preceding node. This is because that there are quite a few nodes in the workflows whose occurrence is not based on its immediately preceding node. FlowRecommender is also superior to the method that performs recommendation using the immediately preceding continuous upstream sub-paths. After a closer examination, we find there are some nodes that correlates with the upstream sub-paths that are not connected with itself.

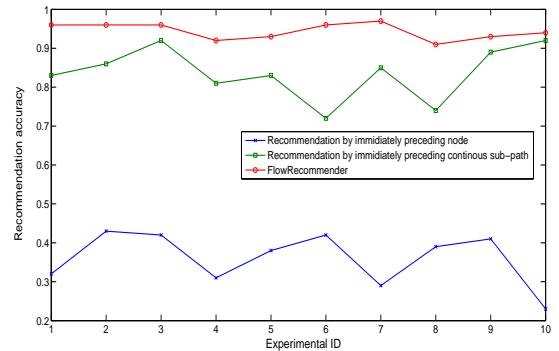


Figure 4: The accuracy of FlowRecommender

3.2 Efficiency Study

We also want to evaluate the efficiency of recommendation. We mainly investigate the execution time of pattern extraction and workflow recommendation which are performed in the offline and online fashion, respectively. Figure 5 and 6 report the execution time of these two steps under varying number of workflows in the provenance. First, from Figure 5, we can see that the extraction of patterns from the provenance scale in an approximately linear manner with respect to the size of the provenance. This is because that the complexity of constructing the inverted indexing of workflows dominates that the step of pattern extraction and, when constructing the indexing, the whole

workflow provenance needs to be scanned. Interestingly, we do not observe such a linear scalability behavior for the workflow recommendation step. As Figure 6 reveals, the execution time of the workflow recommendation step is independent of the provenance size. This is because that it is the pattern table, instead of the original workflow provenance, that needs to be scanned to find the matched patterns for the current workflow under construction. The size of the pattern table is determined by the number of candidate nodes for the whole provenance, i.e., $CNS(D)$. $CNS(D)$ will remain unchanged if the workflows in the provenance are constructed using only a fixed set of nodes. The fluctuation of the execution time results from the different size of $CNS(w)$ at different recommendation locations within the workflow.

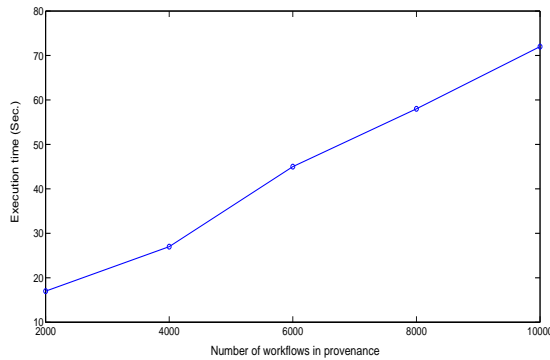


Figure 5: The efficiency of pattern extraction

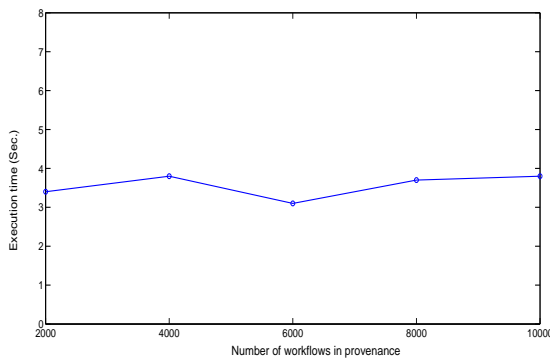


Figure 6: The efficiency of recommendation

4 Related Work

There are abundant research work carried out on service discovery and optimization for composition (which can be considered workflows) in the Service Oriented Computing (SOC) domain. The nodes (in this case, Web services) that are appropriate to the task that are to be fulfilled are first discovered and the best Web service(s) is then identified for execution in the workflow through some service optimization process [3]. In a sense, this resembles to a workflow recommendation problem where the service discovery and optimization help recommend to users the best service that needs to be execute in each step. Nevertheless, there are some fundamental difference between these work and the problem that we are trying to address in this paper. The Web services are recommended based on some pre-defined template where the high-level abstract Web services are well-specified by users when the workflow is constructed. Based upon those abstract Web services, the

so-called concrete Web services (namely the instances of abstract Web services) are discovered and the best one is recommended to construct the workflow. In contrast, the construction of workflows in our problem is purely driven by the workflows themselves archived in the provenance without leveraging any pre-defined templates.

A recommendation service that aims at suggesting frequent combinations of scientific programs for reuse is proposed in [1]. This is an early effort to provide workflow recommendation using provenance. This recommendation service is designed to work over repository of workflow execution logs. It allows users discover useful workflow components and how they can be combined, and collected provenance histories are used to recommend a set of candidate services that may be useful to individual scientists. The drawbacks of this method, however, are as follows:

1. The recommendation of a node in the workflow is only depended on its immediately upstream node. For example, node b is recommended for execution after node a (i.e., $a \rightarrow b$) as long as there exists a high correlation between a and b . However, in many cases, correlation exists for non-consecutive nodes and, therefore, this method is not able to identify these patterns for recommendation;
2. This method performs this on-the-fly in the workflow construction process. As such, this method is not efficient to generate recommendation as the computation of confidence typically involves costly workflow scans.

A more sophisticated workflow recommendation technique is propose in [2]. This technique is designed only as a module in a workflow visualization system, called *vis-Complete*. This method decomposes the original workflow (pipeline) into a number of linear paths with varying number of orders, and the confidence of each possible continuous sub-path in the workflow are quantified in order to provide recommendation. All the possible sub-paths (with varying orders) that terminates at the same node in the workflow are ranked based on the their corresponding confidence score. The downstream node/path that features the highest confidence amongst all candidates is picked up for completing the current workflow. The major drawbacks of this method are summarized as follows:

1. In this method, the recommendation of a node for completing the sub-workflow under construction is dependent on the confidence level of only the paths are immediately connected with this node. For example, given a sub-workflow in the provenance $a \rightarrow b \rightarrow c$, this method will, for node c , evaluate the confidence of c given both $a \rightarrow b$ and b i.e., $Conf(c|a \rightarrow b)$ and $Conf(c|b)$. However, if the actual influencing upstream sub-path structure of node c is node a , then this method is not able to find this pattern for recommendation purpose;
2. This method evaluates the confidence of *all* possible paths which involving calculating the support (i.e., frequency) of both upstream and downstream sub-paths. Given the potentially large number of workflows accumulated in the repository, such calculation is rather expensive.

5 Conclusions and Future Research Directions

In this paper, we propose a new workflow recommendation technique, called FlowRecommender, that leverages provenance of workflows to provide recommendation for the best node (e.g., tool/service/program) that needs to be chosen to complete the workflow. FlowRecommender is able to find the influencing upstream sub-paths of nodes that are not necessarily immediately adjacent to

them. FlowRecommender performs offline pattern extraction step which are maintained in the compact pattern table. This contributes to a highly efficient online recommendation when it is required.

There are some further research directions we are interested in exploring, including

1. First, FlowRecommender is only able to find the most adjacent influencing upstream sub-paths for candidate nodes in its current implementation. It is possible that, however, there exists multiple influencing upstream sub-paths for the same candidate node. The recommendation will fail if the influencing upstream sub-paths other than the one registered in the pattern table are present in the workflow;
2. Second, there have been a plenty of techniques on indexing the sequence patterns. We would like to investigate how these techniques can be used in FlowRecommender to index the influencing upstream sub-paths and to what extent the performance boost can be therefore achieved;
3. Finally, the preliminary experimental evaluation that we have performed is only based upon a synthetic workflow provenance. We plan to utilize the Web service workflow construction system we have developed for biological *in-silico* experiments to collect real-life workflows for a further performance validation of FlowRecommender.

References

- [1] Frederico T. de Oliveira, Leonardo Gresta Paulino Murta, Claudia Werner, Marta Mattoso. Using Provenance to Improve Workflow Design. *2008 International Provenance and Annotation Workshop (IPAW)*, 136-143, 2008.
- [2] David Koop, Carlos Eduardo Scheidegger, Steven P. Callahan, Juliana Freire, Claudio T. Silva. Vis-Complete: Automating Suggestions for Visualization Pipelines. *IEEE Transactions on Visualization and Computer Graphics*, 14(6): 1691-1698, 2008.
- [3] U. S. Manikrao, T.V. Prabhakar, Dynamic Selection of Web Services with Recommendation System, *International Conference on Next Generation Web Services Practices*, 2005.
- [4] Yogesh L. Simmhan, Beth Plale, Dennis Gannon. A Survey of Data Provenance Techniques. *Technical Report*, Computer Science Department, Indiana University, IUB-CS-TR618, 2005.
- [5] C. E. Scheidegger, H. T. Vo, D. Koop, J. Freire, and C. T. Silva. Querying and creating visualizations by analogy. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of Visualization)*, 13(6):1560-1567, 2007.
- [6] D. P. Lanter. Design Of A Lineage-Based Meta-Data Base For GIS. *Cartography and Geographic Information Systems*, vol. 18, pp. 255-261, 1991.
- [7] S. Miles, P. Groth, M. Branco, and L. Moreau. The requirements of recording and using provenance in e-Science experiments. *Technical Report*, Electronics and Computer Science, University of Southampton, 2005.
- [8] I. T. Foster, J.-S. Vockler, M. Wilde, and Y. Zhao. Chimera: A Virtual Data System for Representing, Querying, and Automating Data Derivation. *SS-DBM*, pp. 37-46, 2002.
- [9] D. P. Lanter. Lineage in GIS: The Problem and a Solution. *Technical Report*, National Center for Geographic Information and Analysis, 1990.
- [10] J. Myers, C. Pancerella, C. Lansing, K. Schuchardt, and B. Didier. Multi-Scale Science, Supporting Emerging Practice with Semantically Derived Provenance. *Workshop on Semantic Web Technologies for Searching and Retrieving Scientific Data*, 2003.
- [11] P. Groth, M. Luck, and L. Moreau. A protocol for recording provenance in service-oriented Grids. *OPODIS*, Grenoble, France, 2004.
- [12] R. D. Stevens, A. J. Robinson, and C. A. Goble. myGrid: personalised Bioinformatics on the information grid. *Bioinformatics*, vol. 19, pp. 302i-304, 2003.
- [13] C. Pancerella, etc. Metadata in the collaboratory for multi-scale chemical science. *Dublin Core Conference*, 2003.
- [14] J. Frew and R. Bose. Earth System Science Workbench: A Data Management Infrastructure for Earth-Science Products. *SSDBM*, pp. 180-189, 2001.
- [15] A. Aiken, J. Chen, M. Stonebraker, and A. Woodruff. Tioga-2: A Direct Manipulation Database Visualization Environment. *ICDE*, 1996.

Negative-GSP: An Efficient Method for Mining Negative Sequential Patterns

Zhigang Zheng

Yanchang Zhao

Ziye Zuo

Longbing Cao

Data Sciences & Knowledge Discovery Research Lab
 Centre for Quantum Computation and Intelligent Systems
 Faculty of Engineering & IT, University of Technology, Sydney, Australia
 Email: zgzheng, yczhao, zyzuo, lbcao @it.uts.edu.au

Abstract

Different from traditional positive sequential pattern mining, negative sequential pattern mining considers both positive and negative relationships between items. Negative sequential pattern mining doesn't necessarily follow the Apriori principle, and the searching space is much larger than positive pattern mining. Giving definitions and some constraints of negative sequential patterns, this paper proposes a new method for mining negative sequential patterns, called Negative-GSP. Negative-GSP can find negative sequential patterns effectively and efficiently by joining and pruning, and extensive experimental results show the efficiency of the method.

Keywords: Negative Sequential Pattern, Sequence Mining, Data Mining

1 Introduction

The concept of discovering sequential patterns was firstly introduced in 1995 (Agrawal et al. 1995), aiming at discovering frequent subsequences as patterns in a sequence database, given a user-specified minimum support threshold. Some popular algorithms on sequential pattern mining include AprioriAll (Agrawal et al. 1995), GSP (Generalized Sequential Patterns) (Srikant et al. 1998) and PrefixSpan (Pei et al. 2004). GSP and AprioriAll are both Apriori-like methods based on breadth-first search, while PrefixSpan is based on depth-first search. Some other methods such as SPADE (Sequential Pattern Discovery using Equivalence classes) (Zaki 2001) and SPAM (Sequential Pattern Mining) (Jay et al. 2002) are also widely used in researches.

Different from traditional positive sequential patterns, negative sequential patterns focus on negative relationship between itemsets, in which case, absent items are taken into consideration. We give a simple example to illustrate the differences: Suppose $p_1 = \langle a b c d f \rangle$ is a positive sequential pattern; $p_2 = \langle a b \neg c e f \rangle$ is a negative sequential pattern; and each item a, b, c , etc, stands for a medical item code in the customer claim database of a private health care insurance company. By getting pattern p_1 , we can tell that an insurant usually claimed for a, b, c, d and f in a row; but with pattern p_2 , we are also able to find that given an insurant claim for medical items a and b , and the customer does NOT claim c , he/she would claim item e instead of d later. This kind of

patterns can't be described or discovered by positive sequential patterns like p_1 .

However, while we tried to apply traditional frequent pattern mining algorithm to the negative patterns, two problems stand in the way:

1. Apriori principle doesn't apply to negative sequential patterns. Apriori principle can be simply described as: a sequence can not be frequent if any of its sub-sequences is not. The Apriori principle is widely adopted to reduce the candidate subsequences in positive patterns (Agrawal et al. 1995, Srikant et al. 1998), but it is not necessarily true with patterns containing negative items. Take $c_1 = \langle b \neg c \rangle$, $c_2 = \langle b \neg c a \rangle$ as two candidate patterns and $s = \langle b d a c \rangle$ as sequence data. We can see that s supports c_2 but doesn't support c_1 , which is to say, the pattern c_2 may have greater support than c_1 although c_2 has one more element. We are going to discuss this problem thoroughly in Section 3.

2. Due to the vast candidate space, how can we find frequent patterns efficiently and effectively? Take a 3-length sequence $\langle a b c \rangle$ for instance, it can only support positive candidate patterns $\langle a \rangle$, $\langle b \rangle$, $\langle c \rangle$, $\langle a b \rangle$, $\langle a c \rangle$, $\langle b c \rangle$ and $\langle a b c \rangle$. But in the negative case, sequence $\langle a b c \rangle$ is not only matched with the above positive patterns, but also can be matched with a large bunch of negative candidates, such as $\langle a \neg a \rangle$, $\langle b \neg a \rangle$, $\langle b \neg b \rangle$, $\langle a \neg a c \rangle$, $\langle a \neg c c \rangle$ etc, which makes the searching space much larger.

In this paper, we propose a new method, Negative-GSP, for mining negative sequential patterns based on the GSP algorithm. We also improve the joining and pruning steps for negative sequences to ensure search space integrality and reduce the number of negative candidates as well. Our experiments show that our method can find negative sequential patterns effectively.

2 Related Work

Before negative sequential pattern mining was proposed, a couple of methods have been designed to find negative association rules (Savasere et al. 1998, Wu et al. 2004, Antonie et al. 2004) and negative sequential rules (Zhao et al. 2008). Most early researches on sequential patterns focused on positive relationships, and in recent years, a few researches start to focus on negative sequential pattern mining. The following are some researches pressed in recent years.

Ouyang & Huang (Ouyang et al. 2007) extended traditional sequential pattern definition (a, b) to include negative element like $(\neg a, b)$, $(a, \neg b)$ and $(\neg a, \neg b)$. They put forward an algorithm which finds both frequent and infrequent sequences and then obtains negative sequential patterns from infrequent sequences. A drawback of the algorithm is that a large amount of space is required in order to find both frequent and infrequent sequences.

Nancy et al.(Nance et al. 2007) designed an algorithm named PNSPM (Positive and Negative sequential pattern mining) for mining negative sequential patterns. They applied the Apriori principle to prune redundant candidates, and extracted meaningful negative sequences using the interestingness measure. According to their pattern definition, all elements must be positive except for the last one.

Ouyang et al.(Ouyang et al. 2008) presented the definitions of three types of negative fuzzy sequential patterns: $(a, \neg b)$, $(\neg a, b)$ and $(\neg a, \neg b)$, and then described a method for mining native fuzzy sequential patterns from quantitative valued transactions.

The GSP algorithm(Srikant et al. 1998) is a classical and widely recognized algorithm for sequential pattern mining. GSP makes multiple passes over the dataset to generate frequent sequential patterns. The first pass starts from calculating the support of each single item. At the end of the first pass, all of the 1-item patterns are obtained, which are then used as seeds to generate new candidates for the next pass. Each new candidate has one more item than its seed. The candidates are then pruned to remove infrequent ones. After pruning, the supports of the new candidates are counted by another pass over the dataset and frequent patterns become the seeds for the next pass. The algorithm terminates when there are no frequent patterns at the end of a pass, or when no candidates are generated.

Sue-Chen et al.(Sue-Chen et al. 2008) proposed an algorithm called PNSP (Positive and Negative sequential pattern mining). They presented more comprehensive definitions about negative sequential patterns and extended GSP algorithm to deal with mining negative sequential patterns. Two concepts called *n-cover* and *n-contain* are employed to guide the method. It was claimed that if a *n-cover* value of a candidate is less than the *min-support*, any of its super-sequence is not going to be frequent so the searching of candidate is ended. However, we found out it may cause a loss of some candidates, and so we proposed new measurement to generate and prune candidates.

3 Problem Statement

3.1 Negative Sequence

Definition 1: Sequence

A sequence s is an ordered list of elements, $s = \langle e_1 e_2 \dots e_n \rangle$, where each e_i , $1 \leq i \leq n$, is an element. An element e_i ($1 \leq i \leq k$) includes one or more items. For example, sequence $\langle a b (c, d) e \rangle$ include 4 elements and (c, d) is an element which includes two items.

The length of a sequence is the number of items in the sequence. A sequence with k items is called a *k-sequence* or *k-item sequence*.

Definition 2: Positive/Negative Sequence

A sequence $s = \langle e_1 e_2 \dots e_n \rangle$ is a positive sequence, when each element e_i , $1 \leq i \leq n$ is a positive element. A sequence $s = \langle e_1 e_2 \dots e_n \rangle$ is a negative sequence, when $\exists i$, $1 \leq i \leq n$, e_i is a negative element representing the absence of an element. For example, $\neg c$ and $\neg(c, d)$ are negative elements, so $\langle a b \neg c f \rangle$ and $\langle a b \neg(c, d) f \rangle$ are both negative sequences.

Definition 3: Subsequence

A sequence $s_r = \langle r_1 r_2 \dots r_m \rangle$ is a subsequence of another sequence $s_p = \langle p_1 p_2 \dots p_n \rangle$, if there exists $1 \leq i_1 \leq i_2 \leq \dots \leq i_k \leq n$, $r_1 \subseteq p_{i_1}$, $r_2 \subseteq p_{i_2}$, ..., $r_k \subseteq p_{i_k}$.

Definition 4: Maximum Positive Subsequence

A sequence s_r is a maximum positive subsequence of another sequence s_p , if s_r is a subsequence of s_p , and s_r includes all positive elements of s_p . For example, $\langle a b f \rangle$ is maximum positive subsequence of $\langle a$

Table 1: Examples of pattern matching

Pattern	base-match	match	Sequence
$\langle b \neg c a \rangle$	✓	✓	$\langle b d a \rangle$
$\langle b \neg c a \rangle$	✓	✓	$\langle b d a c \rangle$
$\langle b \neg c a \rangle$	✓	×	$\langle b d c a \rangle$

$b \neg c f \rangle$ and $\langle a b \neg(c, d) f \rangle$.

Definition 5: Negative Sequential Pattern

If the support of a negative sequence is greater than a pre-defined support threshold *min-sup*, and it meets the following constraints, then we call it a negative sequential pattern.

1) Items in a single element should be all positive or all negative. For example, $\langle a (a, \neg b) c \rangle$ is not allowed because item a and item $\neg b$ are in a same element;

2) Two or more continuous negative elements are not accepted in the negative sequence. This constraint is also used by other researchers(Sue-Chen et al. 2008).

3) For each negative item in a negative pattern, its corresponding positive item is required to be frequent. For example, if $\langle \neg c \rangle$ is a negative item, its corresponding positive item $\langle c \rangle$ is required to be frequent.

3.2 Negative Pattern Matching

In order to calculate the support of a negative sequential pattern against the sequence data in database, we need clarify pattern-sequence matching method and criterion. We need to define which patterns can a sequence support.

Definition 6: Negative Base-matching

A negative sequence $s_n = \langle e_1 e_2 \dots e_k \rangle$ *base-matches* a data sequence $s = \langle d_1 d_2 \dots d_m \rangle$, iff, for every negative element e_i , there exist integers p, q, r ($p < q < r$) such that the two conditions hold:

(1) s contains the maximum positive subsequence of s_n ,

(2) $\exists e_{i-1} \subseteq d_p$ and $e_{i+1} \subseteq d_r$, and $\exists d_q$, $e_i \not\subseteq d_q$

That is, each positive element in s_n matches with the same elements in s with same order, while each negative element of s_n can find a match element in s at the corresponding position.

For example, $s_n = \langle b \neg c a \rangle$, it *base-matches* $\langle b d a \rangle$, and also *base-matches* $\langle b d c a \rangle$, since the element d , which matches $\neg c$, can be found between the element b and a .

If a sequence *base-match* a pattern, then we should count it in *base_ssupport* value. We use *base-match* to find seed sequences, which will ensure the integrity of result patterns.

Definition 7: Negative Matching

A negative sequence $s_n = \langle e_1 e_2 \dots e_k \rangle$ *matches* a data sequence $s = \langle d_1 d_2 \dots d_m \rangle$, iff, for every negative element e_i , there exist integers p, q, r ($p < q < r$) such that the two conditions hold:

(1) s contains the maximum positive subsequence of s_n ,

(2) $\exists e_{i-1} \subseteq d_p$ and $e_{i+1} \subseteq d_r$, and $\forall d_q$, $e_i \not\subseteq d_q$

For example, $s_n = \langle b \neg c a \rangle$ *matches* $\langle b d a c \rangle$, but does not match $\langle b d c a \rangle$, since the negative element c appears between the element b and a in s_n .

The above match definition is same as *n-cover* concept in(Sue-Chen et al. 2008). (Sue-Chen et al. 2008) also defined *n-contain*, which is a more restricted match criterion.

A pattern-sequence matching example is given in Table 1. Based on the matching method we have, Apriori-property is not suitable for this case, as we

Table 2: Examples for explain Apriori-Principle

Pattern	match	Sequence
$s_{n1} = \langle b \neg c a \rangle$	×	$\langle b f d c a \rangle$
$s_{n2} = \langle b \neg c d a \rangle$	✓	$\langle b f d c a \rangle$

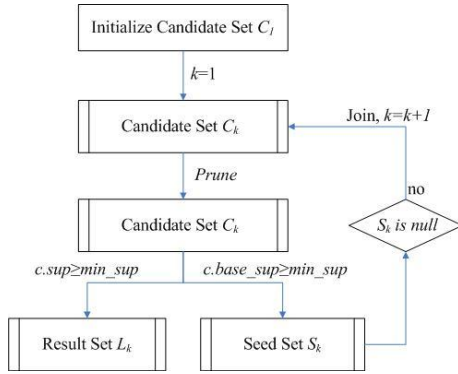


Figure 1: Process Flow of Negative-GSP

pointed out in Section 1. Table 2 gives out a straightforward example. $s_{n1} = \langle b \neg c a \rangle$ and $s_{n2} = \langle b \neg c d a \rangle$ are two candidate patterns and $\langle b f d c a \rangle$ is a sequence in the dataset. The table clearly shows that s_{n1} does not match s , while s_{n2} matches s even if s_{n1} is a sub-sequence of s_{n2} .

Property: Negative Frequent Pattern Property

Based on the above definitions, apparently we have a property: if a negative sequence s is frequent, all its positive subsequence must be frequent; or if the maximum positive subsequence of a sequence s is not frequent, s can not be frequent.

4 Negative-GSP Algorithm

4.1 The Idea

Assume in a sequence data set $D_s = \{d_1, d_2, d_3, \dots, d_n\}$, where d_i ($1 \leq i \leq n$) is a sequence, and our objective is to find frequent negative sequential patterns in D_s , with a minimum support threshold min_sup . The basic process flow as Fig. 1.

First, we utilize the GSP algorithm to generate all positive sequential patterns. Assume $L_{pos} = \{L_{pos,1}, L_{pos,2}, \dots, L_{pos,l}\}$, where $L_{pos,i}$ represents the i -item frequent positive sequential pattern set.

Next, we begin to generate negative sequential patterns based on L_{pos} . We transform all 1-item positive sequential patterns $L_{pos,1}$ to their corresponding 1-item negative sequences, which are taken as the initial 1-item candidates set $C_{neg,1}$. Then we also need prune unnecessary candidates from candidates set $C_{neg,1}$ before calculating the support of every candidates by passing over D_s , which will be discussed in Section 4.3.

Because Apriori Principle doesn't work for negative sequence, the 1-item candidates need to be divided into two classes: one is valid to join with other candidates to generate 2-item negative candidates, and the other is invalid. To verify whether they are valid in the next pass, we use an *base-match* method to verify it (refer to Section 4.4). After this step, we get a processed 1-item seed set $S_{neg,1}$, which is then used as seed to generate a 2-item candidate set $C_{neg,2}$. The candidates with support higher than min_sup are outputted to 1-item frequent patterns $L_{neg,1}$.

Based on the k -item seed set $S_{neg,k}$, we join them with the joining method presented in Section 4.2 to produce a $(k+1)$ -item candidates set $C_{neg,k+1}$. The new candidates may include many invalid candidates,

so pruning invalid candidates is necessary and helpful for further search. Our idea is to verify whether the maximum positive subsequence of the candidate is frequent. Invalid candidates are pruned, and valid ones are kept in the seed set. Then, by passing over the data set D_s , we get the supports of all $(k+1)$ -item candidates. Again, the $(k+1)$ -item frequent candidates with support higher than min_sup are outputted to $L_{neg,k+1}$ as $(k+1)$ -item frequent patterns.

After the above operation, we get the $(k+1)$ -item frequent result $L_{neg,k+1}$ and $(k+1)$ -item seed set $S_{neg,k+1}$ for next pass. Longer patterns are generated by repeating the above process until the candidate set becomes empty. Each iteration will generate and output frequent negative sequential patterns into $L_{neg} = \{L_{neg,1}, L_{neg,2}, \dots, L_{neg,l}\}$, which is the final set of negative sequential patterns.

4.2 Joining to Generate Candidates

The $(k+1)$ -item candidates are generated by joining all k -item seed sequences. Given two k -item seed sequences $c_i = \langle e_{i_1} e_{i_2} e_{i_3} \dots e_{i_{k-1}} e_{i_k} \rangle$ and $c_j = \langle e_{j_1} e_{j_2} e_{j_3} \dots e_{j_{k-1}} e_{j_k} \rangle$, assume $c_i' = \langle e_{i_1} e_{i_2} e_{i_3} \dots e_{i_{k-1}} \rangle$, $c_j' = \langle e_{j_2} e_{j_3} \dots e_{j_{k-1}} e_{j_k} \rangle$. If $c_i' = c_j'$, c_i and c_j are joined to get a $(k+1)$ -item candidate: $\langle e_{j_1} e_{i_1} e_{i_2} \dots e_{i_{k-1}} e_{i_k} \rangle$.

The above joining method is similar to but different from the joining method of GSP. The GSP algorithm gets all $(k+1)$ -item candidates by joining k -item frequent sequential patterns since positive sequences obey the Apriori principle. However, when we mine negative sequential patterns, we need to join not only k -item frequent patterns but also some infrequent k -item sequences, as demonstrated by Section 4.3.

Another point is that, while performing the joining operation, we ignore the joining of positive patterns with themselves, since they have been done in the positive sequential pattern mining at the first step of our algorithm.

4.3 Pruning Unnecessary Candidates

Since the candidates of negative sequential patterns generated during joining step are much more than positive sequential patterns, it is necessary to design an effective pruning method for negative sequential pattern mining. This step prunes some unnecessary candidates from candidates set.

With the GSP algorithm, while pruning in k -item candidates, it prunes all the candidates whose $(k-1)$ -subsequences are not frequent. However, the above pruning method does not work for negative sequential pattern mining in the following ways.

Firstly, given a candidate $C = \langle a b \neg c d \neg e \rangle$. Note that we don't allow two continuous negative elements in a sequential pattern, so $C' = \langle a b \neg c \neg e \rangle$ is an invalid negative sequential pattern. As a result, C will be pruned. However, in fact, the candidate C may be a valid candidate of negative sequential pattern. Secondly, C may be frequent even when its subsequence is not frequent. So we can't prune C simply even its subsequence is infrequent.

Our pruning method is described as follows. For a candidate $C = \langle e_1 e_2 e_3 \dots e_n \rangle$, suppose $C' = \langle e_i e_j \dots e_k \rangle$ is the maximum positive subsequence of C . If C' is not frequent, C must be infrequent and should be pruned. This method is simple but effective to prune invalid candidates without cutting off possible valid candidates by mistake.

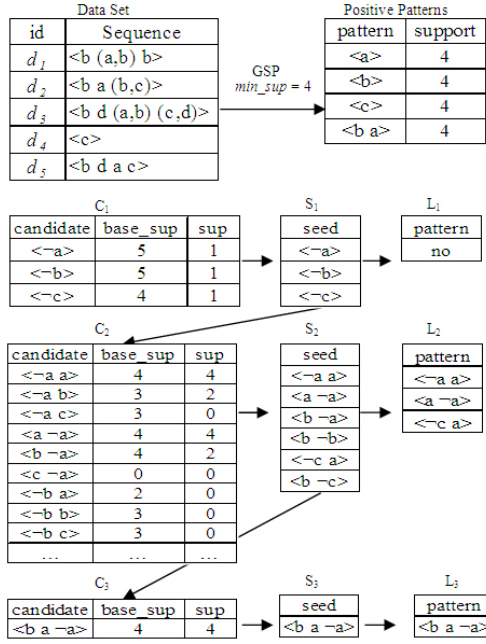


Figure 2: An example

4.4 Generating Seed Set for Next Pass

This step keeps all necessary k -item seed sequences in seed set for next pass, and then $(k+1)$ -item candidates are generated by joining both frequent and infrequent k -item seed sequences.

Given an infrequent 2-item sequence $\langle b -c \rangle$, 3-item candidate $\langle b -c d \rangle$ may still be frequent. So we need count $\langle b -c \rangle$ as seed sequence for joining and generate 3-item candidate $\langle b -c d \rangle$.

It is proposed that the k -item sequences will be regarded as k -item seed sequences if their *base-supports* are greater than min_sup . So we check each candidate's *base-support* value to see whether it is greater than min_sup . If not, it means that the sequence is impossible to generate $(k+1)$ -item frequent pattern and we don't need count it in the seed set again.

4.5 Algorithm Description

Our proposed algorithm for negative sequential pattern mining is given below.

Step 1: Find all positive sequential patterns by the traditional GSP algorithm(Srikant et al. 1998).

Step 2: Transform 1-item positive patterns to 1-item negative patterns as candidates set, and then get 1-item seed set and 1-item patterns.

Step 3: Use all $(k-1)$ -item seeds, perform the joining operation with each other and get k -item candidates and also join $(k-1)$ -item positive patterns with $(k-1)$ -item seed sequences.

Step 4: Prune unnecessary candidates to get a smaller candidate set.

Step 5: Count all candidates' supports and base-supports.

Step 6: If the base-support is greater than min_sup , then the candidate is added to k -item seed set. If the support is greater than min_sup , then the candidate is frequent and is outputted as a k -item pattern.

Step 7: If k -item seed set is not empty, increase k by one and loop back to Step 3 until the next candidate set is empty.

The procedure is illustrated with an example in Fig 2.

Table 3: Synthetic datasets

Parameters	DS1	DS2
Number of sequences (DB)	10k	100k
Number of items (N)	1k	10k
Average number of elements per sequence (C)	8	10
Average number of items per element (T)	8	2.5
Average length of maximal potentially large sequences (S)	4	4
Average size of itemsets in maximal potentially large sequences (I)	8	2.5

Negative-GSP Algorithm

Input: Dataset DS , min_sup

Output: Negative patterns L

```

/* S: Seed set;
   C: Candidate set;
   L: Result set; */

k = 1;
Ck = initialize();
Sk = Ck;
while ( Sk.size() > 0 ){
    Ck+1 = Join(Sk);
    Ck+1 = Prune(Ck+1);
    for ( each c in Ck+1 ){
        if ( c.basesupport > min_sup ) Sk+1.add(c);
        if ( c.support > min_sup ) Lk+1.add(c);
    }
    L.add(Lk+1);
    k = k + 1;
}
return L;

```

5 Experiments and Results

5.1 Datasets

Two synthetic datasets generated by IBM data generator(Agrawal et al. 1995) are used to test our algorithm in the experiments.

Dataset1(DS1) is C8.T8.S4.I8.DB10k.N1k, which contains 10k sequences and the number of items is 1000, the average number of elements in a sequence is 8, the average number of items in an element is 8, average length of maximal pattern consists of 4 elements and each element is composed of 8 items averagely.

Dataset2(DS2) is C10.T2.5.S4.I2.5.DB100k.N10k, which contains 100k sequences and the number of items is 10k, the average number of elements in a sequence is 10, the average number of items in an element is 2.5, average length of maximal pattern consists of 4 elements and each element is composed of 2.5 items averagely.

5.2 Experiments Results

We compared the runtime of negative sequential pattern mining with different support thresholds (see Fig 3), and also compared counts of patterns with different support thresholds (see Fig 4). Negative pattern mining spends much more runtime than positive pattern mining because the candidates counts are not of the same magnitude, especially when the support threshold is set very low.

5.3 Comparison with PNSP Algorithm

We compared our method with PNSP algorithm(Sue-Chen et al. 2008). The results of runtime (see Fig 5)

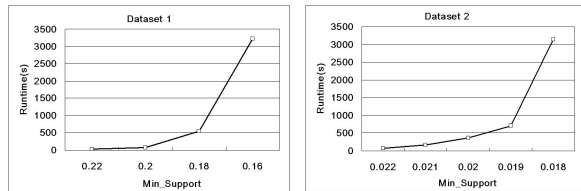


Figure 3: Runtime on DS1 and DS2

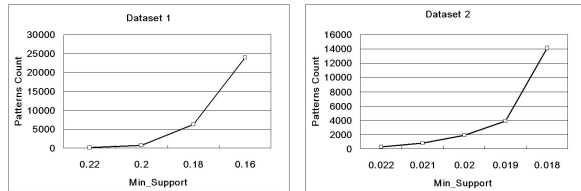


Figure 4: Patterns counts on DS1 and DS2

show that our method outperforms PNSP. The reason is that PNSP generates more unnecessary candidates. Especially when the number of negative frequent items increased, its negative candidates may increase sharply. For our algorithm, when there are a lot of negative candidates, it will cost much running time in the joining process for new candidates. So when *min_sup* is very low, Negative-GSP can't get very good performance as when *min_sup* is high.

6 Conclusions and Future Work

In this paper, we presented the definitions for negative sequential patterns, and proposed a method for negative sequential pattern mining based on the GSP algorithm. We also designed effective pruning method to reduce the number of candidates. The efficiency and effectiveness of our algorithm is shown in our experiments on two synthetic datasets.

In our future research, we will focus on selecting interesting rules from the discovered negative sequential patterns. How to find interesting and interpretable rules from a lot of negative sequential patterns is valuable in business applications. Another research topic is to find more effective pruning method that can reduce candidates more effectively and avoid unnecessary candidates.

Acknowledgements

This work was supported by the Australia Research Council (ARC) Linkage Project LP0775041 and Discovery Projects DP0667060 & DP0773412, and by the Early Career Researcher Grant from University of Technology, Sydney, Australia. The author of this paper acknowledge the financial support of the Capital Markets CRC.

References

Agrawal, R. & Srikant, R. (1995), Mining Sequential Patterns. In: Yu, P.S., Chen, A.L.P. (eds.): 11th

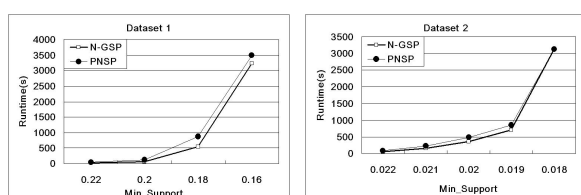


Figure 5: Comparison with PNSP Algorithm

International Conference on Data Engineering. I E E E, Computer Soc Press, Taipei, Taiwan, pp. 3–14

Antonie, M.L. & Zaiane, O.R. (2004), Mining positive and negative association rules: An approach for confined rules. Proceedings of the 15th European Conference on Machine Learning/8th European Conference on Principles and Practice of Knowledge Discovery in Databases. Springer-Verlag Berlin, Pisa, ITALY, pp. 27–38

Jay, A., Jason, F., Johannes, G. & Tomi, Y. (2002), Sequential Pattern mining using a bitmap representation. Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, Edmonton, Alberta, Canada

Nancy, P.L., Hung-Jen, C. & Wei-Hua, H. (2007), Mining negative sequential patterns. Proceedings of the 6th Conference on WSEAS International Conference on Applied Computer Science - Volume 6. World Scientific and Engineering Academy and Society (WSEAS), Hangzhou, China

Ouyang, W., Huang, Q. & Luo, S. (2008), Mining Positive and Negative Fuzzy Sequential Patterns in Large Transaction Databases. Fuzzy Systems and Knowledge Discovery, 2008. FSKD '08. Fifth International Conference on, Vol. 5, pp. 18–23

Ouyang, W.M. & Huang, Q.H. (2007), Mining Negative Sequential Patterns in Transaction Databases. Machine Learning and Cybernetics, 2007 International Conference on, Vol. 2, pp. 830–834

Pei, J., Han, J., Mortazavi-Asl, B., Jianyong, W., Pinto, H., Qiming, C., Dayal, U. & Mei-Chun, H. (2004), Mining sequential patterns by pattern-growth: the PrefixSpan approach. Knowledge and Data Engineering, IEEE Transactions on 16, pp. 1424–1440

Savasere, A., Omiecinski, E. & Navathe, S. (1998), Mining for strong negative associations in a large database of customer transactions. 14th International Conference on Data Engineering, Proceedings, pp. 494–502

Srikant, R. & A., R. (1998), Mining sequential patterns: Generalisations and performance improvements. Proceedings of the Fifth International Conference on Extending Database Technology (EDBT)

Sue-Chen, H., Ming-Yen, L. & Chien-Liang, C. (2008), Mining Negative Sequential Patterns for E-commerce Recommendations. Proceedings of the 2008 IEEE Asia-Pacific Services Computing Conference. IEEE Computer Society

Wu, X.D., Zhang, C.Q. & Zhang, S.C. (2004), Efficient mining of both positive and negative association rules. ACM Trans. Inf. Syst. 22, pp. 381–405

Zaki, M.J. (2001), SPADE: An efficient algorithm for mining frequent sequences. Machine Learning 42, pp. 31–60

Zhao, Y., Zhang, H., Cao, L., Zhang, C. & Bohlscheid, H. (2008), Efficient Mining of Event-Oriented Negative Sequential Rules. Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT '08. IEEE/WIC/ACM International Conference on, Vol. 1, pp. 336–342

Non-Redundant Rare Itemset Generation

Yun Sing Koh¹

Russel Pears²

School of Computing Science and Mathematics
Auckland University of Technology, New Zealand,
Email: ykoh@aut.ac.nz¹, rpears@aut.ac.nz²

Abstract

Rare itemsets are likely to be of great interest because they often relate to high-impact transactions which may give rise to rules of great practical significance. Research into the rare association rule mining problem has gained momentum in the recent past. In this paper, we propose a novel approach that captures such rare rules while ensuring that redundant rules are eliminated. Extensive testing on real-world datasets from the UCI repository confirm that our approach outperforms both the Apriori-Inverse (Koh et al. 2006) and Relative Support (Yun et al. 2003) algorithms.

Keywords: Rare Association Rule Mining, Apriori-Inverse, Non-Redundant Itemset

1 Introduction

Association rule mining (Agrawal et al. 1993) is used to find common or frequent patterns within datasets. In the classical association rule mining process, all frequent itemsets are found, where an itemset is said to be frequent if it appears above a minimum frequency threshold s , called minimum support. Association rules are then derived from frequent items and are represented in the form $A \rightarrow B$ where AB is a frequent itemset. Strong association rules are those that meet the minimum confidence c threshold (the percentage of transactions containing A that also contain B).

The minimum support threshold is used as a noise filter to eliminate itemsets that do not appear often within the dataset. This threshold has to be sufficiently strong to reduce frequent itemsets to a manageable level. However, in some data mining applications relatively infrequent associations are likely to be of great interest as they relate to rare but crucial cases. Application domains that benefit from rare association mining include the diagnosis of rare diseases, the prediction of telecommunication equipment failure, and the identification of associations between infrequently purchased supermarket items.

For example in a supermarket transactional dataset, most purchasing behavior follows a very regular and predictable pattern, which is related to daily household items such as bread, butter, and milk. However, there also exists behavior which is uncharacteristic in respect to the volume of items sold

when compared to the staple items mentioned earlier. Such behavioral patterns are potentially useful to the retailer as they could involve associations between items that are highly profitable. However, because of the relative infrequency with which such associations manifest, traditional frequent association mining techniques would be unable to capture the patterns involved. This is due to the combinatorial explosion in the number of candidate itemsets generated by setting the minimum support to a low enough value to capture the rare associations. Such an explosion in the search space renders traditional algorithm such as Apriori unusable. This problem was first highlighted by Cohen et al. (2000), showing that association between expensive items such as vodka and caviar are likely to be infrequent but interesting due to their high value. There are numerous techniques which tries to solve this problem (Liu et al. 1999, Szathmary et al. 2007, Yun et al. 2003, Koh et al. 2006, Koh & Pears 2008, Taniar et al. 2008).

Current rare itemset generation techniques suffer from three issues. Firstly, their rule generators produce a mix of both frequent and infrequent rules (Liu et al. 1999, Szathmary et al. 2007). Furthermore, the rule bases they produce contain rules that could be inferred from other rules, thus making them redundant. In our experimentation we noticed the occurrence of such redundant rules when we did a comparative analysis with the Apriori-Inverse rule generator. Thirdly, they generate rules having only infrequent items in their rule terms, which represents only a sub class of rare rules (Koh et al. 2006). Such rule generators do not produce rare rules that consist of frequent items in their rule terms. Such rare rules are valuable as they represent scenarios where individual rule terms in rule antecedents are frequent on their own but rare in combination with each other. Such rules capture very specific but hard to detect events in their rule consequents, on account of their rarity. Up until now there has been no effective solution to the problems referred to earlier and this research represents an attempt to address each of these issues.

In this paper we introduce a novel approach called Non-Redundant Itemset Generation (N-RIG) which seeks to capture rare patterns by using an efficient pruning strategy, without the need for pre-processing the dataset by partitioning it. The non-redundant generator ensures that only itemsets that lead to an improvement in rule prediction accuracy are ever considered for rule generation. Pruning of redundant items is achieved by the introduction of a constraint that we introduce, known as Cumulative Productive Confidence (CPC).

The remainder of the paper is organized as follows. Section 2 provides a review of related research in the area of rare association rule mining. In Section 3 we give a brief overview of our new approach to finding rare association rules. In Section 4 we discuss

Copyright ©2009, Australian Computer Society, Inc. This paper appeared at the Eighth Australasian Data Mining Conference (AusDM 2009), Melbourne, Australia. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 101, Paul J. Kennedy, Kok-Leong Ong and Peter Christen, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

the redundant itemset problem. In Section 5 we introduce the notion of an adaptive support threshold which further enhances the quality of the rules produced. Experimental results of applying the method on several real-world datasets is presented in Section 6. The paper concludes in Section 7 with a summary of the contributions made in this research.

2 Related Work

The efficient detection of rare association rules with low support but high confidence is a difficult data mining problem. To find such rules with traditional approaches such as the Apriori would require the minimum support (minsup) threshold to be set very low, resulting in large computational overhead while producing a large rule base, parts of which contain redundant rules. As a specific example of the problem, consider the association mining problem where we want to determine if there is an association between buying a food processor and buying a cooking pan (Liu et al. 1999). The problem is that both items are rarely purchased in a supermarket. Thus, even if the two items are almost always purchased together when either one of them is purchased, this association may not be found. Modifying the minsup threshold to take into account the importance of the items is one way to ensure that rare items remain in consideration. To find this association minsup must be set low. However setting this threshold low would cause a combinatorial explosion in the number of itemsets generated. Frequently occurring items will be associated with one another in an enormous number of ways simply because the items are so common that they cannot help but appear together. This is known as the rare item problem (Liu et al. 1999). It means that the application of Apriori-like approaches are unlikely to yield rules that indicate rare events of potentially dramatic consequence.

Liu et al. (1999) note that some individual items can have such low support that they cannot contribute to rules generated by Apriori, even though they may participate in rules that have very high confidence. They overcome this problem with a technique called MSAPriori whereby each item in the database can have a minimum item support (MIS) given by the user. By tailoring the MIS value for different items, a higher minimum support is tolerated for rules that involve frequent items and a lower minimum support for rules that involve less frequent items. Yun et al. (2003) proposed the RSAA algorithm to generate rules in which significant rare itemsets take part, without the need for any user specified thresholds. This technique uses relative support measure, RSup in place of support. The RSup measure serves to decrease the support threshold for items that have low frequency and to increase the support threshold for items that have high frequency. In common with Apriori and MSAPriori, RSAA is exhaustive in its generation of rules, and will generate rules which are not rare (i.e. rules with high support and high confidence).

Szathmary et al. (2007) presented an approach for rare itemset mining from a dataset that splits the problem into two tasks. The first task, the traversal of the frequent zone in the space, is addressed by two different algorithms, a naive one, Apriori-Rare, which relies on Apriori and hence enumerates all frequent itemsets; and MRG-Exp, which limits the considerations to frequent generators only. They consider computation of the rare itemsets that approaches them starting from the bottom of the itemset lattice and then moving upwards through the frequent zone. They defined a positive and the negative border

of the frequent itemsets, and a negative lower border and the positive lower border of the rare itemsets, respectively. An itemset is a maximal frequent itemset (MFI) if it is frequent but all its proper supersets are rare. An itemset is a minimal rare itemset (mRI) if it is rare but all its proper subsets are frequent. If the minimum-allowable relative support value is set close to zero, MRG-Exp takes a similar amount of time to that taken by Apriori to generate low-support rules due to the need for sifting through the high-support rules.

Koh et al. (2006) proposed an approach to find rare rules with candidate itemsets that fall below a maxsup (maximum support) level but above a minimum absolute support value. They introduced an algorithm called Apriori-Inverse to find sporadic rules efficiently: for example, a rare association of two common symptoms indicating a rare disease. They used a maximum support threshold to prune out any items that may be frequent. They then use a minimum absolute support (minabssup) threshold value derived from an inverted Fisher's exact test (Weinstein 2005) to prune out noise. At the low levels of co-occurrences of candidate itemsets that need to be evaluated to generate rare rules, there is a possibility that such co-occurrences happen purely by chance and are not statistically significant. The Fisher test provided a statistically rigorous method of evaluating significance of co-occurrences and was thus an integral part of their approach. The main drawback of this method is that it cannot detect rare rules that embed frequent items in their rule terms.

Koh & Pears (2008) proposed a pre-processing mechanism, based on transaction clustering to generate rare association rules. The basic concept underlying transaction clustering stems from the concept of large items as defined by traditional association rule mining algorithms. In their approach, they partition the dataset and then run the Apriori-Inverse algorithm on each of the clusters found. They showed that pre-processing the dataset by clustering improves rule quality by as each cluster is able to express its own associations without interference or contamination from other sub groupings that have different patterns of relationship. The rare rules produced by each cluster were shown to be more informative than the rare rules found from direct association rule mining on the original unpartitioned dataset.

We have based our approach on Apriori-Inverse. Thus we use the minabssup threshold based on Fisher's exact test to filter out chance co-occurrences. However we differ from Apriori-Inverse in that we use the maxsup threshold only in the rule generation phase and not the candidate itemset phase. The rationale for this is explained in Section 4. In the next three sections we present our approach for rare association rule generation.

3 Our Approach

This section presents the key concepts governing the Non-Redundant Rare Itemset Generation (N-RIG) approach. The focus of our approach is to find rare rules that contain rule terms that may by themselves be frequent whilst preventing the generation of redundant rules. Our approach is adapted from the Apriori algorithm. Similar to Apriori, our approach is set in two phases, the candidate generation and the rule generation phase. We discuss the candidate generation phase below. The candidate generation phase itself consists of two steps.

In the first step, we allow itemsets that are above a minabssup threshold which we adopt from (Koh et al. 2006) and which fulfil our Cumulative Produc-

tive Confidence (CPC) measure to be extended. The minabssup threshold is calculated for every itemset and is used to eliminate noise and is used instead of a fixed minimum support threshold. The CPC measure is used to eliminate *redundant* itemsets. We define a redundant itemset(I) as one that gives rise to a rule that can be inferred from a rule covered by some subset of itemset I. In the next section, we discuss the CPC measure in detail.

In the second step, use a maximum support threshold to prune the set candidate itemsets further. Here we prune out all itemsets that have support above the maximum support threshold. This is needed as we are only interested in rare itemsets.

Algorithm 1 N-RIG algorithm

Input: Transaction Database D , universe of items I , maximum support (maxsup) value
Output: Non-redundant Rare Itemsets
 $N \leftarrow |D|$
 $k \leftarrow 1$
 $R_k \leftarrow \{\{i\} | i \in \text{dom } Idx, \text{count}(\{i\}) > 1\}$
while $R_k \neq \emptyset$ **do**
 $k \leftarrow k + 1$
 $C_k \leftarrow \{x \cup y | x, y \in R_{k-1}, |x \cap y| = k - 2\}$
 $R_k \leftarrow \{c | c \in C_k, \text{supp}(c) > \text{minabssup}, \text{CPC}(c) > 0\}$
end while
 $R_k \leftarrow \{c | c \in R_k, \text{supp}(c) < \text{maxsup}\}$
return $\bigcup_{t=2}^{k-1} R_t$

4 The Redundant Itemset Problem

Despite the fact that Apriori-Inverse outperforms its rivals on performance and rule quality, there exists two areas where its performance can be improved. Firstly, it is possible that Apriori-Inverse generates rules that are redundant. In its itemset generation phase Apriori-Inverse combines itemsets A and B as long as they pass the Fisher test (Weisstein 2005). While the Fisher test does an excellent job of filtering out itemsets that co-occur together by chance, it does not guarantee rule minimality in the rule base that it generates.

Consider the following example with a dataset containing 50 transactions. Suppose that we have 3 itemsets A, B and C with support 20, 30 and 25 respectively. If $\text{supp}(AB) = 18$ and if AB co-occurs with every transaction with C , then we have $\text{supp}(AC) = 18$. With these statistics the Fisher test determines that items A and B do not occur by coincidence, thus Apriori-Inverse will record itemset AB as a candidate itemset for rule generation. If the minimum confidence threshold is set to 0.8 then the rule $A \rightarrow B$ will be generated as the rule confidence at $18/20 = 0.9$ exceeds the confidence threshold set.

Since $\text{supp}(AC) = 18$ it follows that this itemset too will pass the Fisher test, thus producing AC as another itemset. In the next level of itemset generation Apriori-Inverse will consider the generation of ABC from the candidate pairs AB and AC . Now $\text{supp}(ABC) = \text{supp}(AB)$ since A always co-occurs with C and hence it follows that ABC will also pass the Fisher test. This in turn leads to the following rule:

$AC \rightarrow B$ This rule too meets the confidence threshold as its confidence is:

$$\frac{\text{supp}(ABC)}{\text{supp}(AC)} = \frac{\text{supp}(AB)}{\text{supp}(AC)} = 1$$

since $\text{supp}(AB) = \text{supp}(AC)$.

However, it is clear that Rule 2 is redundant in the presence of Rule 1. Rule 1 captures the minimal conditions required to predict the occurrence of B given A . This example illustrates that Apriori-Inverse is

vulnerable to the redundant rule generation problem. Whilst it is possible to apply a post rule generation filter to remove such redundant rules, a more efficient approach would ensure that itemsets such as ABC are never generated in the first place. The generation of itemset ABC has the potential to lead to even more redundancy as all pairs of itemsets such as (ABC, ABD) with a common prefix of AB propagates the redundancy of ABC with other items such as D , leading to many more itemsets such as $ABCD$ that give rise to redundant rules. This is clearly undesirable since the candidate generation phase is the performance bottleneck in the association rule mining process.

Our approach avoids this problem by pruning itemsets such as ABC from the set of candidates, thus ensuring that redundancy is eliminated at its source. We use an improvement measure called *CPC*, that ensures that any given itemset will only be extended if its extension produces an increase in the *CPC* measure over the improvement value when the itemset itself was being formed. In section 4.1 we show that no itemset that passes the improvement test will be redundant.

The second issue with Apriori-Inverse is that it uses a fixed threshold for determining rarity. The use of a fixed threshold inhibits the discovery of rules for items whose support is above the threshold but who co-occur together with support less than the threshold set. Consider for example items X and Y with support 0.2 and 0.3 respectively. Suppose that the support of XY is 0.08, and the maximum support threshold is set at 0.10. Apriori Inverse only combines items that meet the maximum support threshold constraint and thus X and Y will not be combined together, although their combination gives rise to a rare rule with support 0.08. This simple example illustrates that it would be desirable to explore items in the neighborhood of the maximum support threshold with a view to expanding Apriori-Inverse's rule base to capture rare rules that contain one or more terms that are frequent. Such types of rules are of interest in many types of applications. Such applications include disease diagnosis where certain symptoms occur on their own commonly but whose co-occurrence points to a specific disease that occurs rarely in the population.

We now turn our attention to the issue of preventing the occurrence of redundant rules.

4.1 Redundancy Removal

As mentioned above Apriori-Inverse is vulnerable to the problem of redundant rules. Such redundant rules contain terms in the rule antecedent that do not contribute to an increase in the rule confidence. Such rules not only increase the size of the rule base unnecessarily, but also tend to mislead the decision maker into thinking that certain terms need to be satisfied in the antecedent when in reality they do not.

We first give a formal definition of rule redundancy from Bayardo (Bayardo 1998). Consider a generic rule $X \rightarrow Y$. The improvement, I , of such a rule is:

$$I(X \rightarrow Y) = \text{conf}(X \rightarrow Y) - (\text{conf}(Z \rightarrow Y)), \max(Z \subset X)$$

A redundant rule can now be defined as one whose improvement is less than zero. We prevent the occurrence of such rules by defining a metric called *Cumulative Productive Confidence* (CPC) that measures whether an extension to a given itemset will ensure that all rules that can be produced as a result of the extension have greater confidence than the rules

produced with the original non extended version the itemset.

Suppose that we have itemsets X and Y that have passed the Fisher test. Itemset X will be merged with Y and extended to XY if it satisfies the condition below:

$$CPC(XY) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X)} - \frac{\text{supp}(Y)}{\arg \min_{W \subset Y} \text{supp}(W)} > 0$$

Theorem 1 below offers a formal proof that the CPC measure inhibits the production of redundant rules.

Theorem 1. *All rules produced from an extension of an itemset that satisfies the CPC constraint defined above will be non redundant.*

Proof. Itemsets X and Y to be merged need to have a common prefix so we will represent X as AB and Y as AC . We now have $X \cup Y = ABC$. For ABC to be a legitimate itemset, itemset $Z = BC$ must also exist and have passed the Fisher test, since ABC can also be produced by $Y \cup Z$ and thus we cannot have ABC without Z passing the Fisher test as well. In order to produce ABC , it then follows that we must have

$$CPC(X, Y) > 0 \quad (1)$$

$$CPC(X, Z) > 0 \quad (2)$$

$$CPC(Y, Z) > 0 \quad (3)$$

From 1 we have:

$$\frac{\text{supp}(AB \cup AC)}{\text{supp}(AB)} - \frac{\text{supp}(AC)}{\arg \min_{W \subset AC} \text{supp}(W)} > 0$$

This implies that:

$$\frac{\text{supp}(ABC)}{\text{supp}(AB)} - \frac{\text{supp}(AC)}{\text{supp}(A)} > 0 \quad (4)$$

since

$$\frac{\text{supp}(AC)}{\text{supp}(A)} \geq \frac{\text{supp}(AC)}{\arg \min_{W \subset AC} \text{supp}(W)}$$

From 4 it follows that

$$\text{conf}(AB \rightarrow C) > \text{conf}(A \rightarrow C)$$

By substituting C instead of A in 4 above we also have: $\text{conf}(AB \rightarrow C) > \text{conf}(C \rightarrow A)$. Thus the rule $AB \rightarrow C$ is non redundant. From 2 we have:

$$\frac{\text{supp}(BC \cup AC)}{\text{supp}(BC)} - \frac{\text{supp}(AC)}{\arg \min_{W \subset AC} \text{supp}(W)} > 0$$

From this we derive:

$$\frac{\text{supp}(BCA)}{\text{supp}(BC)} - \frac{\text{supp}(AC)}{\text{supp}(A)} > 0 \quad (5)$$

As with 4 above we have:

$$\frac{\text{supp}(AC)}{\text{supp}(A)} \geq \frac{\text{supp}(AC)}{\arg \min_{W \subset AC} \text{supp}(W)}$$

From 5 it follows that:

$$\text{conf}(BC \rightarrow A) > \text{conf}(A \rightarrow C)$$

By substituting C instead of A in 5 above we also have:

$$\text{conf}(BC \rightarrow C) > (\text{conf}(C \rightarrow A))$$

We thus have the rule $BC \rightarrow C$ non redundant and lastly using 3 above we have:

$$\frac{\text{supp}(AC \cup BC)}{\text{supp}(AC)} - \frac{\text{supp}(BC)}{\arg \min_{W \subset AC} \text{supp}(W)} > 0$$

which leads to: $\text{conf}(AC \rightarrow B) > \text{conf}(A \rightarrow C)$ and $\text{conf}(AC \rightarrow B) > \text{conf}(C \rightarrow A)$ which means that rule $BC \rightarrow C$ is also redundant. We have thus shown that all rules produced by the extension are non redundant and this proves the theorem.

5 Adaptive Thresholding

Although N-RIG dispenses with a maximum support threshold during itemset generation it still uses such a threshold during the rule generation phase to ensure that only rare rules are generated. However, the use of such a threshold can have undesirable effects if its value is set arbitrarily. For example, by setting a threshold at 0.10 on a dense dataset, we would be letting through more itemsets when compared to setting the threshold at the same value on a sparse dataset. To find a suitable cut off point we use an adaptive threshold based on a modified version of a hill climbing algorithm. We inspect the support of the candidate itemset. Using a list of itemsets sorted in ascending order of support, we compare the support of itemset x to the support of itemset $x + 1$. If the difference of the support is less than $k\%$, the new support threshold is set as $\text{supp}(x + 1)$. The process is repeated until the difference between two consecutive itemsets is more than $k\%$, and we consider that we have reached a partition in the itemset support distribution that defines a suitable threshold value.

In the next section, we present the results from our approach and compare them with those produced by the Apriori-Inverse algorithm.

6 Evaluation and Results

In this section, we compare the performances of the standard Apriori-Inverse and RSAA algorithms with our proposed algorithm. The experiments were performed on a Windows Vista machine with Intel Duo Core having 3.0GHz CPU and 2.68 GB of RAM. Testing of the algorithms was carried out on 5 different datasets from the UCI Machine Learning Repository (Newman et al. 1998). Table 1 represents the summary of the results found using Apriori-Inverse, N-RIG, and RSAA algorithms. For Apriori-Inverse and N-RIG we set the maximum support threshold (max-sup) to 0.10 for all datasets. In all of the experiments, we set the minimum confidence threshold to 0.90. For a comparison have reported the number itemsets found by RSAA which fell below the 0.10 threshold.

We compare the time taken to produce the rare itemsets. Overall our approach generated more itemsets when compared to Apriori-Inverse. On the average, we generated 1229 itemsets as compared to Apriori-Inverse which generated an average of 682. The N-RIG approach is not merely confined to generating itemsets that contain only infrequent items, unlike Apriori-Inverse. This explains the difference in the overall number of itemsets produced between the two approaches. Despite the greater effort expended by N-RIG in expanding the scope of rare itemsets produced its run time compares well with that of Apriori-Inverse. The number of rare itemsets produced by RSAA was consistently higher than with the other two algorithms. In line with the greater number of itemsets produced RSAA runtime were also much

Table 1: Summary of Experimental Results

Dataset	Apriori-Inverse		N-RIG		RSAA	
	Rare Itemset	Time (s)	Rare Itemset	Time (s)	Rare Itemset	Time (s)
Flag	72	0.51	260	7.60	1210	98.2
Hepatitis	31	0.08	51	2.42	398	19.53
Soybean-Large	135	0.45	1289	88.16	6226	2388.67
Audiology	123	0.51	239	12.30	N/A	N/A
Mushroom	3051	55.76	4305	349.00	5804	360.56

higher than with the other two algorithms. In the case of the Soybean dataset, RSAA performed very poorly with respect to the runtime. As for the Audiology dataset, RSAA did not terminate after two hours and hence we decided to exclude it from the comparison.

6.1 Comparative Analysis

Table 2 shows clearly that both Apriori-Inverse and N-RIG both produce rules with high lift with the top 20 Lift values being identical for the larger datasets, Soybean and Mushroom. However, the lift values for RSAA was significantly smaller for these datasets. For the two smaller datasets, Flag and Hepatitis Apriori-Inverse generated just 3 and 7 rules respectively and so a meaningful comparison with N-RIG was not possible. RSAA produced mixed results for two smaller datasets, giving a higher lift for Hepatitis while producing a lower lift value for the Flag dataset.

Table 2: Rule Lift across selected UCI datasets

Dataset	Apriori-Inverse	N-RIG	RSAA
Flag	-	12.3	5.5
Hepatitis	-	6.70	11.0
Audiology	100	34.7	N/A
Soybean-Large	51.2	51.2	15.4
Mushroom	1015.5	1015.5	6.3

Table 3 illustrates a clear difference in behavior between the algorithms. While the top 20 rule support and overall rule support values are broadly similar for the Apriori-Inverse and N-RIG algorithms, the rule term support for N-RIG was significantly higher, particularly for the Soybean and Mushroom datasets. As shown in Table 3 the average antecedent rule support for the Mushroom dataset at 2.6% is a factor of 26 times higher than the corresponding value for Apriori Inverse. The same trend holds true for the smaller datasets, albeit on a smaller scale. We chose to exclude RSAA from further analysis as its lift values were smaller than with the other two approaches.

These results suggests that N-RIG is better able to capture rare rules where individual terms are frequent. As pointed out in Section 1 such rules are of great practical significance.

Such rules manifest with N-RIG as it is not restricted by a maximum support constraint in its candidate itemset generation phase, unlike Apriori-Inverse, thus enabling the former to produce rule terms of higher support than the latter. We next examine some of the rules discovered by N-RIG which Apriori-Inverse was unable to generate.

6.1.1 Mushroom Dataset

N-RIG produced a number of very rare and very high lift rules involving different combinations of the same terms appearing together. One such rule is given below with support 0.1% and Lift of 1015.5. Such extremely rare rules in the occurrence of a relatively

Table 3: Rule Support and Rule Term Support Comparison

Dataset	Apriori-Inverse			
	Support (Top 20 Rules)	Antecedent Rule Support	Consequent Rule Support	Support (Entire Rule Base)
Flag	2.1%	5.2%	2.1%	4.1%
Hepatitis	3.9%	3.9%	3.9%	7.1%
Audiology	1.0%	1.2%	3.3%	1.2%
Soybean-Large	2.0%	2.0%	2.3%	5.2%
Mushroom	0.1%	0.1%	0.3%	0.4%
Dataset	N-RIG			
	Support (Top 20 Rules)	Antecedent Rule Support	Consequent Rule Support	Support (Entire Rule Base)
Flag	2.1%	6.1%	25.4%	5.6%
Hepatitis	3.9%	4.3%	4.3%	16.1%
Audiology	1.0%	2.7%	8.1%	2.7%
Soybean-Large	2.6%	3.9%	22.8%	3.9%
Mushroom	0.3%	2.6%	32.5%	2.5%

dense dataset such as Mushroom tends to boost the Lift value to such heights and the Lift factor taken by itself is not indicative of the rule interestingness. Indeed, the other two rules given below appear to be more interesting, despite the fact that their Lift values are much smaller.

population:c,
stalk-color-above-ring:y
→ stalk-color-below-ring:y
stalk-surface-above-ring:y
veil-color:y

N-RIG was also able to discover subclasses of the two varieties of Mushrooms, namely the edible and poisonous species. Two such rules are given below.

stalk-color-above-ring:n
→ edible:p,
stalk-shape:e,
stalk-surface-below-ring:k

The above rule, with Lift 6.3, is interesting as it covers only 5.3% of the dataset and it applies to a subclass of poisonous mushrooms that cover only 11% of the total poisonous variety.

gill-attachment:a → cap-color:n, edible:e

The above rule (with Lift 5.9) is even more interesting as it covers only 2.3% of the dataset and applies to a subclass of edible mushrooms that cover only 4.9% of the edible variety.

6.1.2 Audiology Dataset

The Audiology dataset also produced some rare rules of interest. Two such examples are given below:

history_dizziness:t
history_fluctuating:t → class:possible_menieres

The above rule, with support 2.2% and Lift 25 identifies a histories of dizziness and fluctuating hearing levels as being strongly associated with a disorder of the inner ear that can affect hearing and balance.

class:conductive_fixation \rightarrow tymp:as
ar_c:absent

This rule, having support 2.6 % and Lift 18.2 indicates that hearing disorder conductive fixation occurs in the absence of a condition coded as “tymp:as ar_c”.

6.1.3 Adaptive Threshold

Table 4: Results for Adaptive Threshold

Dataset	N-RIG with Adaptive Threshold		
	Rare Itemset	Increment from normal N-RIG	Time (s)
Flag	393	39.6%	8.0
Hepatitis	4	-92.1%	2.39
Soybean-Large	2775	115.3%	88.95
Audiology	308	28.9%	12.7
Mushroom	5005	16.2%	351.25

Table 4 displays the effect of using an adaptive threshold with N-RIG. Out of the five datasets tested, four of the datasets produced more rare itemsets when compared to the arbitrary threshold set at 0.10 as given in Table 1. This denotes that the partition was set at a higher level than the 0.10 support value, whereas for one of the datasets the adaptive threshold value did not reach the 0.10 mark. This is due to the fact that the adaptive threshold value is dependent on the dataset that is being analysed.

7 Conclusion

In this research we have shown that the Non Redundant Itemset Generation (N-RIG) approach produces rules of practical significance that could not be discovered efficiently by the two other methods that we compared our approach with. By dispensing with an arbitrary maximum support during the candidate generation phase and replacing it with the Cumulative Productive Confidence measure we were able to generate rare rules with high frequency terms whilst keeping run time down to reasonable bounds.

References

- Agrawal, R., Imielinski, T. & Swami, A. (1993), Mining association rules between sets of items in large databases, in P. Buneman & S. Jajodia, eds, ‘Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data’, pp. 207 – 216.
- Bayardo, R. (1998), Efficiently mining long patterns from databases, in ‘SIGMOD ’98: Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data’, ACM Press, New York, NY, USA, pp. 85–93.
- Cohen, E., Datar, M., Fujiwara, S., Gionis, A., Indyk, P., Ullman, J. D., Yang, C., Motwani, R. & Motwani, R. (2000), Finding interesting associations without support pruning, in ‘ICDE ’00: Proceedings of the 16th International Conference on Data Engineering’, IEEE Computer Society, Washington, DC, USA, p. 489.
- Koh, Y. S. & Pears, R. (2008), Rare association rule mining via transaction clustering, in J. F. Roddick, J. Li, P. Christen & P. J. Kennedy, eds, ‘Seventh Australasian Data Mining Conference (AusDM 2008)’, Vol. 87 of *CRPIT*, ACS, Glenelg, South Australia, pp. 87–94.
- Koh, Y. S., Rountree, N. & O’Keefe, R. (2006), ‘Finding non-coincidental sporadic rules using apriori-inverse’, *International Journal of Data Warehousing and Mining* **2**(2), 38–54.
- Liu, B., Hsu, W. & Ma, Y. (1999), Mining association rules with multiple minimum supports, in ‘Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining’, pp. 337 – 341.
- Newman, D., Hettich, S., Blake, C. & Merz, C. (1998), ‘UCI repository of machine learning databases’, <http://www.ics.uci.edu/~mlearn/MLRepository.html>.
- Szathmary, L., Napoli, A. & Valtchev, P. (2007), Towards rare itemset mining, in ‘ICTAI (1)’, IEEE Computer Society, pp. 305–312.
- Taniar, D., Rahayu, W., Lee, V. & Daly, O. (2008), ‘Exception rules in association rule mining’, *Applied Mathematics and Computation* **205**(2), 735 – 750. Special Issue on Advanced Intelligent Computing Theory and Methodology in Applied Mathematics and Computation.
- Weisstein, E. (2005), ‘Fisher’s exact test’, MathWorld—A Wolfram Web Resource. <http://mathworld.wolfram.com/FishersExactTest.html>.
- Yun, H., Ha, D., Hwang, B. & Ryu, K. H. (2003), ‘Mining association rules on significant rare data using relative support’, *The Journal of Systems and Software* **67**(3), 181 – 191.

Monetising User Generated Content Using Data Mining Techniques

Yu-Hsn Liu¹, Yongli Ren², Robert Dew¹

¹School of Information Technology, Deakin University
221 Burwood Highway, Victoria 3125, Australia
{yuhsnliu, rad}@deakin.edu.au

²School of Information Engineering, Zhengzhou University
Zhengzhou 450052, China
yonglitom@gmail.com

Abstract

Social media systems such as YouTube are gaining phenomenal popularity. As they face increasing pressure and difficulties monetising the large amount of user-generated content, there are intense interests in technologies capable of delivering revenue to the owners. In this paper, we propose to use data mining techniques to help companies increase their revenue stream. Our approach differs principally in the underlying monetisation model and hence, the algorithms and data utilised. Our new model assumes both consumer and commercial content being entirely user-generated. We first present an algorithm to demonstrate one of possible monetisation technique that could be used in social media systems such as YouTube. A large volume of real-data harvested from YouTube will also be discussed and made available for the community to potentially kick start research in this direction.

Keywords: YouTube, User-Generated Content, Monetisation, Web Mining, Data Mining, Business Intelligence.

1 Introduction

Three years ago, most of the content published by the media exists as a linear stream coming from a single information source such as a TV channel, the radio, or the newspaper. The 'consumer' as the name suggests is largely responsible for the consumption of information. Their role in publishing or their influence in the content is minimal in most cases.

As communication technologies improve significantly in speed, capacities and forms, we are seeing the emergence of a new media. Characterised largely by 'user contribution', 'sharing', 'decentralisation' and being 'free', these social media systems are gaining phenomenal popularity and success on the Internet. FaceBook, MySpace, YouTube, Wikipedia, and other Web 2.0 sites are overtaking traditional media and to a certain extent, creating transparency levels never seen before.

Just Australia alone, the significance and impacts are clear. In the last two years, many traditional media reported poor earnings results (Cartman, Australian Media 2007, Australian Associated Press 2007, Becker & Posner 2009), and the fire sales of traditional media (Ali Moore 2009, Australian Associated

Press 2009) only appear to confirm the bearish outlook of these businesses. As more users turn towards a new paradigm of content consumption, where they are also publishers on a collaborative and free platform, the traditional approach of monetising published content needs to be relooked.

Most social media systems operate without boundaries and are unconstrained by geographical locations, language and time differences. Consequently, they have a subscriber base many times larger than most traditional media in existence. YouTube for example generates more than 100 million views a day and receives more than 65,000 video uploads in 24 hours (Cha, Kwak, Rodriguez, Ahn & Moon 2007). This level of consumption and content creation delivered YouTube the video publishing power that traditional media is incapable of matching. Yet, the success of these systems is also the very reason for their poor financial position as their exponential growth result in significant costs that is matched by a disproportional income stream.

YouTube for example is reportedly losing millions of dollars every day (Fritz 2009, Silversmith 2009, Hartley 2009) because viewers get the content for free. From a business perspective, there is no way YouTube could charge viewers a fee no matter how small that is. Similarly, it is not possible to charge the content publishers who are users themselves. YouTube's mantra of keeping content free soon became the reason of its current success and also the threat of its future failure. Therefore, there is an increasing pressure among such content providers to monetise their businesses (Steffens 2009, Dignan, Diaz & Nusca 2009, Bogatin 2009) before investors pull out.

The idea of monetising content is not new. TV channels, radios and newspapers all publish content for 'free' (or a small fee) in return for 'eyeballs' that could be sold to businesses in the form of advertisements. Social media systems simply adopt this model in hope of achieving the same outcome. *The Age* for example inserts commercial footages on the electronic version of their news. This approach annoys a large number of readers as they find the content intrusive, consumes their bandwidth (which they have to pay for), and are most of the time, untargeted.

If these social media systems are to continue operations, a new monetisation model and appropriate mechanisms are needed. In this paper, we propose a new monetisation model based on user-generated content and user meta-data. This model excludes businesses from the direct involvement of the content users consume. Instead, they would identify user-generated content to push commercial messages on their behalf. To achieve this, we believe data mining technologies would be the best candidate. However, existing algorithms will need to be redesigned to utilise the new model so as to bring about significant

Copyright ©2009, Australian Computer Society, Inc. This paper appeared at the Eighth Australasian Data Mining Conference (AusDM 2009), Melbourne, Australia. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 101, Paul J. Kennedy, Kok-Leong Ong and Peter Christen, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

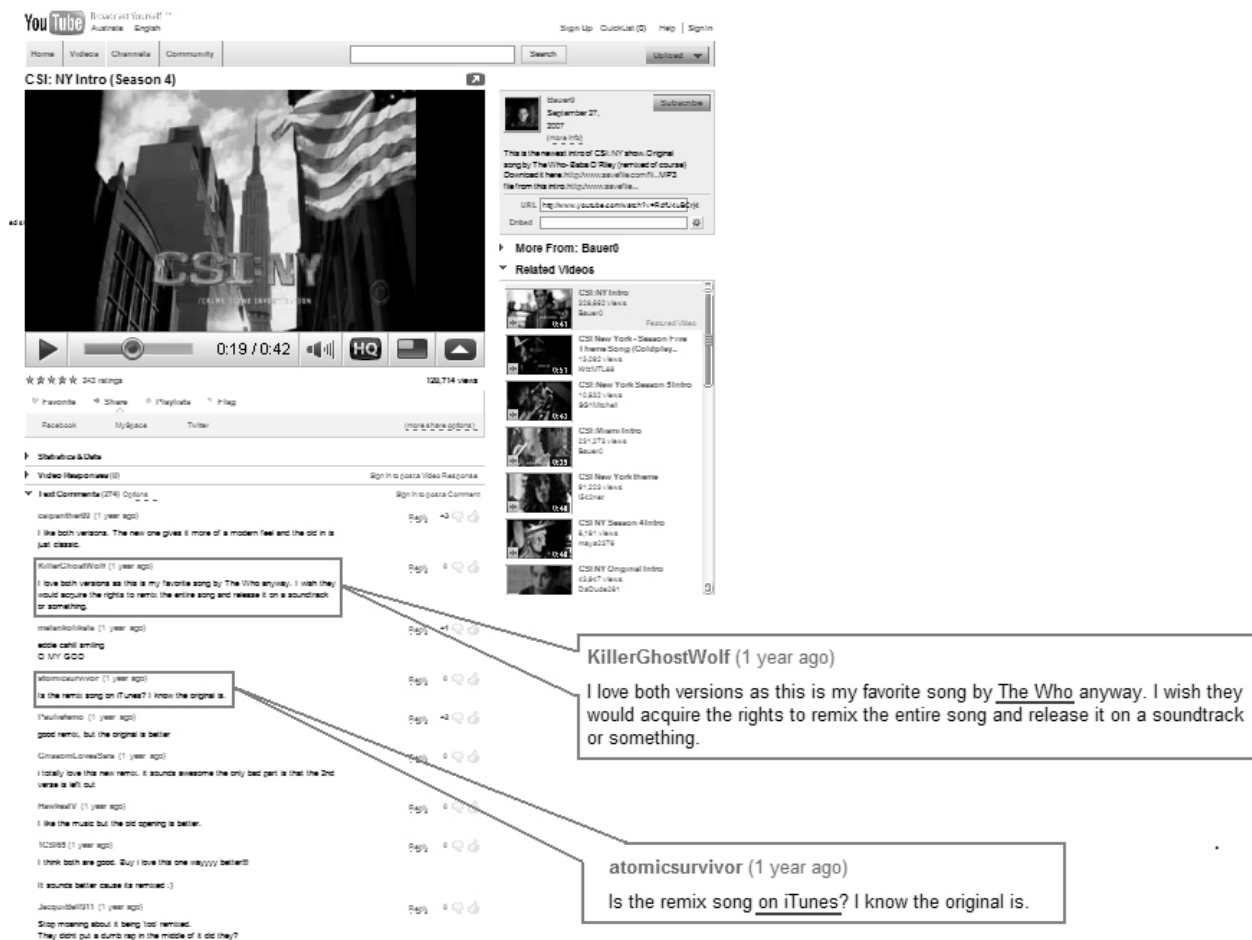


Figure 1: One of the many *CSI: New York* videos posted on YouTube. We will discuss our monetisation model based on this real example. Source: <http://www.youtube.com/watch?v=RdfU4uBCrj4>. Notice that there isn't any advertisements, which means one monetisation opportunity lost. While it's possible to include an advertisement, it is likely to be untargetted for TV programme like *CSI: New York*. Nevertheless, we could exploit what's in the comments, since the comments are a result of watching the video.

increase in the revenue stream.

Among the many social media systems, we will focus on systems similar to YouTube, where the underlying content is user-generated and that user-generated meta-data (e.g., video profiles and video comments) are available. On this particular model, we make the following contributions:

- We propose a novel monetisation model, where both the consumer and commercial content (as well as meta data) are entirely user-generated.
- Using the suggested monetisation model, we proposed a possible monetisation scenario and an algorithm aimed at increasing the revenue streams of content providers.
- As a consequent of undertaking the above research work, we will also contribute our real-world data sets harvested from YouTube so as to provide a platform for other researchers to explore this new direction.

The remaining sections of this paper is organised as follows. In the next section, we discuss further details of our monetisation model. Specifically, we will demonstrate the potential of our proposal with an example. In Section 3, we suggest a monetisation algorithm to realise our proposed scenario. In Section 4, we present our preliminary results before presenting our conclusions in Section 5.

2 A New Monetisation Model

The novelty of our approach lie in the observation that advertisements do not necessarily deliver the same level of impact on users of social media systems than messages delivered by a user within their community. With the bulk of social media users in the age of 20 to 30 years old, they made up a significant group whose beliefs are radically different. According to (McCrindle 2009), this group value collaboration, sharing and the freedom of opinion. As a result, they tend to be more receptive to peer opinions rather than commercial messages.

The 'Gen-Y' group of users aside, chances are that many individuals have seek peer opinions (or are influenced by peer comments) when it comes to making a decision about a product or service. Therefore, the significance of peer opinions cannot be undermined. While it was common to manage user opinions to minimise the level of negativity of an organisation's business, user opinions are now leveraged to improve on the positivity of an organisation's product or service. This is commonly seen in the virtual world demonstrated by two key technologies: collaborative filtering (Herlocker, Konstan, Terveen & Riedl 2004, Herlocker, Konstan, & Riedl 2000) popularised by Amazon.com and viral marketing (Domingos 2005, Leskovec, Adamic & Huberman 2005).

While social media systems have capitalised on these characteristics to fuel their growth, monetisa-

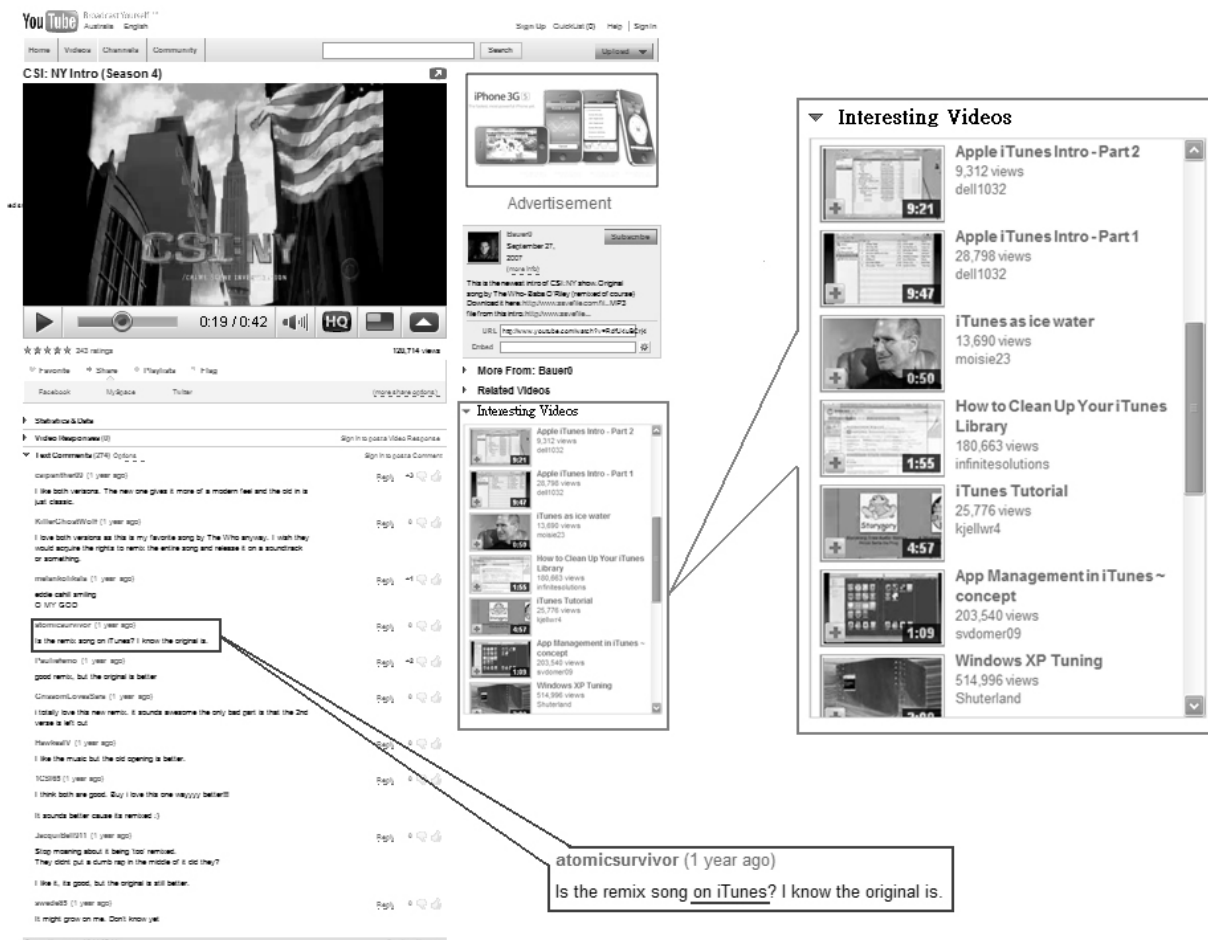


Figure 2: This is a mock up screen showing the enhancements possible, and realising the scenario we discussed in Section 2. In terms of the interface change, this is almost undetectable from the user's perspective. For the media system however, this change together with the monetisation mechanism that works in the background will deliver the revenue stream it needs.

tion methods haven't. Ideally, monetisation methods should exploit the user characteristics the same way the success of social media systems have. More importantly unless an improved revenue stream is achieved, the future of these systems is unclear. Given the large amount of user videos created for the products they use, it is clear that there is a large group of users who would like to offer their views in addition to the commercial messages. Regardless of whether those views are negative or positive, their videos offer a more balanced evaluation for the next potential customer. This led us to consider the possibility of creating a monetisation mechanism, where the commercial messages could be entirely user-generated. To appreciate the virtue of this model, let us consider a possible scenario.

Our example uses YouTube and will assume the readers know how the system operates, and have seen some user comments before. Many of us would have watched one of our favourite TV programmes on YouTube either because you missed the telecast, or you want to watch a certain part again. Let's say you want to watch *CSI: New York* on YouTube as Figure 1 shows. On a system like YouTube not only do you get the video, you would notice a few existing features such as comments posted by other users, a list of related videos found on the right, and also an advertisement along with other features. In this case, there isn't any advertisement shown. This means one monetisation opportunity lost. Even if an advertisement is shown, chances are the advertisement isn't

related to *CSI: New York* and would appear random to most users. At first thought, one may argue that a TV programme like *CSI: New York* do not contain material for target marketing. If they do, TV advertisements would not be the way they are now.

We counter argue that this is not true. In fact, the level of target marketing is increasingly apparent even in conventional TV programmes. *MasterChef, Australia* for example has more food related advertisers than a programme like *Big Brother*. On YouTube, the fact that it has all the user generated comments provide a fertile ground for target advertisements to be taken to a new level. Potentially, what conventional TV programmes cannot do, e.g., target marketing on a generic programme like *CSI: New York*, YouTube could with all the other meta information.

To see how this is possible, let us consider a real life example using a CSI video from YouTube (Figure 1 contains the URL for this video). As mentioned, the user comments is one of the differences between watching the same programme on TV and YouTube. If we are to advertise without intruding the viewer and if our intention is to keep the advertisements targeted, one way is to monitor the comments of the video being watched.

In our example, one of the users commented on the theme song of *CSI: New York*. This sparked a number of related posts with one user eventually mentioning *iTunes* and another mentioning the band who sung the theme song – *The Who*. As a result of these comments, there is a probability that a viewer is influ-

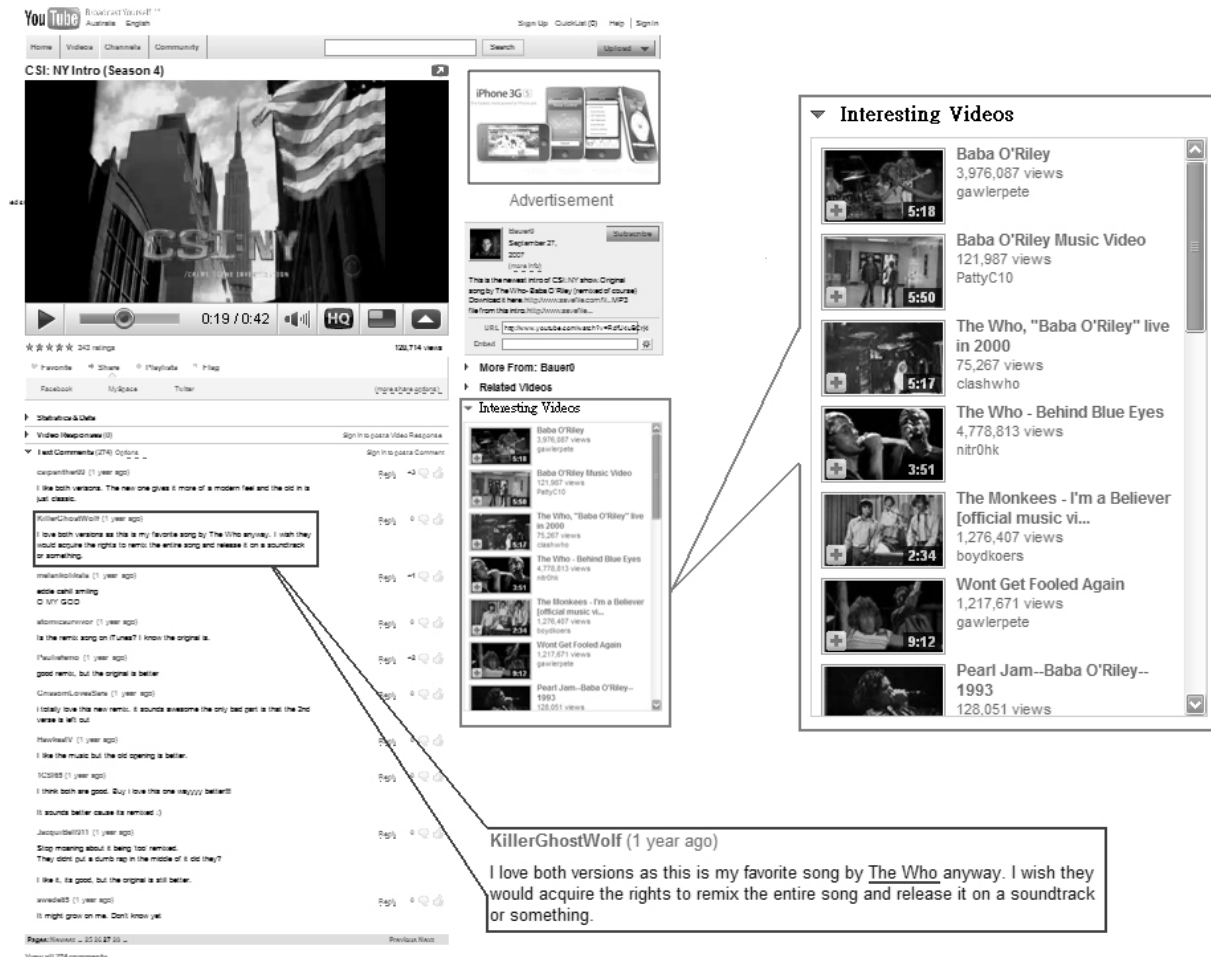


Figure 3: This is another mock up screen showing the effects of our proposed monetisation mechanism. In this case, the mention of both *iTunes* and *The Who* band could be utilised by displaying an advertisement from Apple, and also populating the *Interesting Videos* section with the band videos. Clicking on the band videos may or may not result in direct monetisation. A more ideal system (which we have yet to address in our algorithm) is to direct a click closer towards monetisation. In this case, clicking on the *The Who* videos may not results in direct monetisation but it would maintain the eyeball and eventually perhaps, lead to buying an album by the band.

enced to find out more about *The Who*, or even go online to purchase the song from *iTunes*. Clearly, if the appropriate monetisation mechanisms are in place, an *iTunes* advertisement could be used in place of a random one. This would deliver significant improvements in the click-through. Even better, we could also introduce user-created videos on *iTunes* and videos on *The Who* band along with the CSI video. Figure 2 shows a mock up of the existing screen in Figure 1 to help our readers visualise our monetisation scenario. Notice the *iTunes* advertisement from Apple instead of no advertisement (or a random one) because *iTunes* was detected in the user comments. Additionally, notice a new section called *Interesting Videos* been added after *Related Videos*. While *Related Videos* contains a list of CSI videos, *Interesting Videos* are actually videos that are retrieved from keywords such as *iTunes* and *The Who*, which when clicked could potentially lead to monetisation opportunities if those videos were sponsored by the commercial entities.

From the technical perspective, our interest is in how we could populate the *Interesting Video* section. This is the section where its videos, when clicked, will lead to monetisation. Therefore, an underlying requirement for videos listed in this section is that they must have a monetary value attached. This monetary value could be a payment from the advertiser who selects a user-created content as its agent for commer-

cial messages about a product, or a video that would increase the likelihood of a user clicking on an advertiser's commercial message.

Our example also illustrated the possibility of having more than one keyword in the comments that could lead to multiple sets of videos being candidates of monetisation. In Figure 2, we show how the mention of *iTunes* could lead to targeted advertisements and also a list of *Interesting Videos* about *iTunes*. Another possible scenario from the other keyword is the mention of *The Who* band – see Figure 3. If the commercial owners of the band wants to increase publicity, they could pick some of the user produced *The Who* videos to include in the *Interesting Video* section. Consequently, these videos become monetisable for the social media system but rised another technical challenge – in the presence of multiple candidate keywords (and thus multiple sets of *Interesting Videos* for monetisation), which video should we decide upon?

3 Monetisation Algorithm

Now that we have discussed the possible monetisation scenario, we turn our attention to the discussion of a possible monetisation algorithm.

Let $U = \{u_1, u_2, \dots, u_i\}$ be the set of all users (or

Algorithm 1 CreateInterestingList(user u , video v)

```

1: Let  $C(u, v) = \text{GetComments}(u, v)$ 
2: for all  $c \in C(u, v)$  do
3:    $c' \leftarrow \text{StemWord}(c)$ 
4:   for all  $w \in c'$  do
5:      $\mathcal{C}_w = \{\phi\}$ 
6:     if  $w \in \mathcal{M}$  then
7:        $\mathcal{C}_w = \mathcal{C}_w \cup \{v_t | w \subseteq v_t.\text{MonetisationKeyword}\}$ 
8:     end if
9:   end for
10: end for
11: Let  $\mathcal{R} = \{\phi\}$ 
12: for all  $w \in \{c'_1, c'_2, \dots, c'_j\}$  do
13:   Sort  $\mathcal{C}_w$  such that  $\mathcal{C}_w = \{\mathcal{U}(v_{t1}) \geq \mathcal{U}(v_{t2}) \geq \dots \geq \mathcal{U}(v_{t\ell})\}$ 
14:   Let  $\mathcal{C}'_w = \{\text{top } n\text{th elements of } \mathcal{C}_w\}$ 
15:    $\mathcal{R} = \mathcal{R} \cup \{\mathcal{C}'_w\}$ 
16: end for
17: Sort  $\mathcal{R}$  such that  $\mathcal{R} = \{\sum_{i=1}^{|\mathcal{C}'_{w1}|} \mathcal{U}(v_{ti}) \geq \sum_{i=1}^{|\mathcal{C}'_{w2}|} \mathcal{U}(v_{ti}) \geq \dots \geq \sum_{i=1}^{|\mathcal{C}'_{wj}|} \mathcal{U}(v_{ti})\}$ 
18: return  $\mathcal{C}'_{w1}$  in sorted  $\mathcal{R}$ 

```

user accounts) in the social media system. For a social media system like YouTube, a user u_j can upload a number of videos. We denote the videos uploaded by u_j as $V(u_j) = \{v_1, v_2, \dots, v_k\}$. For a given video v from a user u , a set of comments made about v by other users of the system are available. We will denote this as $C(u, v) = \{c_1, c_2, \dots, c_\ell\}$.

For a user u watching a video v , the objective of our monetisation algorithm is to find a set of videos $\mathcal{T} = \{v_1, v_2, \dots, v_m\}$ such that for every $v_t \in \mathcal{T}$, v_t 's monetisation keyword contains one or more word terms w_1, w_2, \dots found in $C(u, v)$. For convenience, we will use an object-oriented notation when referring to a user, video or comment property. Therefore, a video v_t will have a monetisation keyword denoted as $v_t.\text{MonetisationKeyword}$.

Since the choice of a video can be highly subjective from user to user, it will be difficult to guarantee a click-through. In other words, we may have selected v_t as the video that leads to monetisation but the user may not necessarily click on it. Therefore, given a set of candidate videos \mathcal{C} (or \mathcal{T}), we want to rank (or rate) every $v_t \in \mathcal{C}$ so that we can pick the best video in \mathcal{C} for monetisation. To do this, we define a utility measure $\mathcal{U}(v_t \in \mathcal{C})$ to quantify the probability (or likelihood) of v_t delivering a click-through (Joachims, Granka & Pan 2005, Jansen 2009, Zhao, Liu, Bhowmick & Ma 2006). Clearly, there are many ways one could define utility and is likely to vary even within the same system for users on different geographical location. For discussion sake, we define the utility \mathcal{U} of a video $v_t \in \mathcal{C}$ as

$$\mathcal{U}(v_t) = f_1(v_t.\text{ViewCount}) + f_2(v_t.\text{Rating}) + \dots \quad (1)$$

where f_1 and f_2 are functions that would output a normalised value based on the property of v_t , in this case **ViewCount** and **Rating**, so that a consistent score could be obtained for each video. Finally, a word term w belongs to \mathcal{M} , the collection of monetisable keywords if and only if w is a keyword tagged to a video $v_t \in \mathcal{T}$. In other words, given $w \in \mathcal{M}$, we have a set of videos $\{v_t | w \subseteq v_t.\text{MonetisationKeyword}\}$. Given these definitions, we can now present our monetisation algorithm as shown in Algorithm 1.

For a given user u and video v , the algorithm will retrieve all the comments $C(u, v)$ associated with v using **GetComments()** as shown in Line 1. For each comment c , we will first stem the words so as to make matching easier. While in theory this maybe sufficient, our preliminary analysis of the raw

data highlights a number of technical challenges. Of the 3,480,580 comments investigated, there are large number of short-form words, e.g., 'dun' for 'don't' or 'b4' for 'before', that stemming will not adequately address. At the same time, a lot of symbols need to be removed including ones like 'xoxo', or ':)', which has no direct bearing on monetisation.

Once the entire line of comment has been stemmed (and 'cleaned' of short forms and symbols), we will cycle through each word term w in Line 4. If $w \in \mathcal{M}$ holds, then we will add all the videos $\{v_t | w \subseteq v_t.\text{MonetisationKeyword}\}$ to \mathcal{C}_w making the videos our candidates for monetisation for w as shown from Lines 5 to 9. Once all the word terms are processed, we sort the videos in each \mathcal{C}_w for each w by their utility \mathcal{U} keeping only the highest n^{th} videos from \mathcal{C}_w in \mathcal{C}'_w . This is then added to \mathcal{R} representing the all the candidate videos \mathcal{C}'_{wj} under consideration. We then find the sum of utility for all videos associated with a given word term w as represented in Line 17 – $\sum_{i=1}^{|\mathcal{C}'_{wj}|} \mathcal{U}(v_{ti})$. This is then sorted with the collection of videos having the most utility being selected, i.e., \mathcal{C}'_{w1} in sorted \mathcal{R} (Line 18).

At this point in time, we have not considered the speed issue in favor of an easy way to quickly test our idea. Hence, the algorithm's focus is on the logic of the recommendation rather than the practicality of its implementation within a system such as YouTube. In addition, data cleaning regarding the short-forms and symbols were omitted, which we admit could affect the accuracy of our matching a word term to \mathcal{M} . We also hand-picked word terms that are easily recognisable as keywords for monetisation by counting and looking for word terms referring to a product as the basis for establishing \mathcal{M} and \mathcal{T} in advanced. All these considered, we wish to highlight to the readers the preliminary nature of our model and we concede the need for further investigation before we could reliably report the impacts of this model. On a significantly small sample and controlled test environment however, the results look very promising.

Finally, we conclude this section with an interesting note for the curious reader. The 'iTunes' word term in our example was mentioned 1,117 times from over 3.4 million comments spread across 623,730 videos and 65,645 users. While these numbers appear to be large, we remind our users that YouTube records 65,000 video uploads a day. With 623,730 videos (equivalent to looking at 10 days of video uploads) we are really looking at the tip of an iceberg

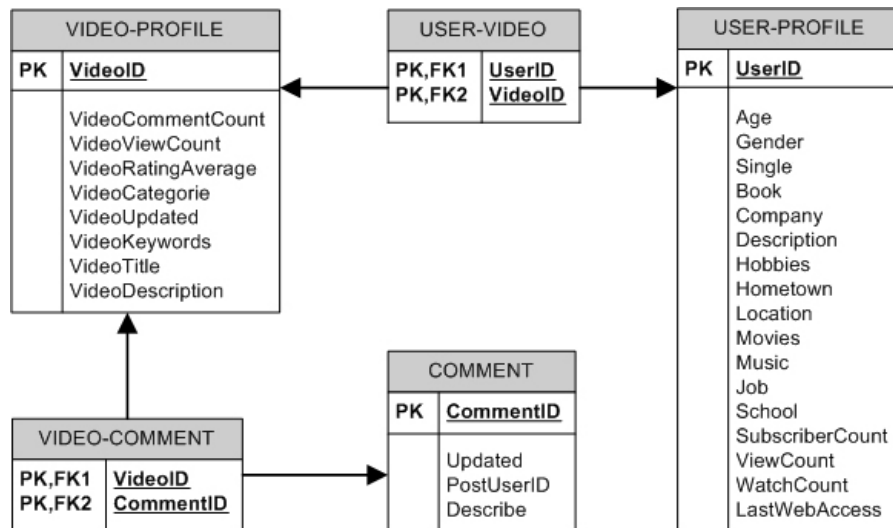


Figure 4: ER-diagram depicting the relationship and structure of our harvested YouTube data. The data can be obtained from our research Webpage at <http://www.deakin.edu.au/~yuhsnliu/youtube>.

and therefore, should not recklessly dismiss its significance.

4 Data Sets

We next present the data we harvested from YouTube. Figure 4 shows the ER diagram depicting the way we structured the information we harvested. From this ER-diagram, we were able to load our data files onto a database system to produce joins of a flat file so that different types of analysis could be carried out. Over the lifetime of this project, we endeavor to release weekly updates of what we harvested from our crawler. As we use SQL Server and wrote our initial code in C# and .NET, instructions on how to load the files onto SQL Server, how to create joins, and how to export the joins as a flat file for analysis are available along with our data sets. Also, the description of each fields and the ER-diagram can be found on our Website. The URL of the Webpage is given in Figure 4.

As a condition of use, the data sets are meant for research and non-profit purposes only. Any form of commercial use should be consulted with the authors. We also kindly request that proper acknowledgement is made when using the data sets downloaded from our Website. Further information on the citation details can be found on the Webpage at the URL given in Figure 4.

5 Conclusions

Social media systems with rich video content are emerging rapidly in recent years. As collaborative access and sharing of information becomes the 'norm', it becomes vital that businesses utilise these systems and incorporate social media technologies in their operations. *The Age* for example may be a news publisher but incorporating social media technologies on their Website allows them to deliver content through a new dimension. In doing so, it is important that monetisation tools are available so that *The Age* can continue to deliver the cutting edge experience to their readers via a sustainable business model.

In this paper, we contribute to the above in three ways. First, we propose to use user-generated con-

tent throughout our monetisation process. We argue the effectiveness of such an approach based on the user characteristics of these social media systems. Second, we propose a monetisation algorithm based on our monetisation model to realise the scenario discussed in Section 2. While our results are preliminary, we are confident that with further refinement to the algorithm and possibly the development of a prototype, it will be possible to demonstrate its impact in the near future. Finally, we will make available the large volume of user-generated content we harvested on YouTube to the research community. With access to this real-world data sets, not only will we help advance existing data mining research but also, we may potentially spark of new research ideas in data mining - particularly, in the area of using data mining to achieve monetisation of user-generated content.

References

- Cartman/Australian Media. Packer Sells Nine to CVC. <http://www.australian-media.com.au/news/4217/packer-sells-nine-to-cvc/>, last accessed on July 31, 2009.
- Australian Associated Press. Ten Tipped to Post Loss as Ads Vanish. <http://business.theage.com.au/business/ten-tipped-to-post-loss-as-ads-vanish-20090401-9jgv.html>, last accessed on July 31, 2009.
- Ali Moore. PBL Considers Further Media Sell-off. <http://www.abc.net.au/lateline/business/items-/200705/s1935762.htm>, last accessed on July 31, 2009.
- Australian Associated Press. PBL Sell-off to Ignite Gaming Expansion. <http://www.theage.com.au/news/Business/PBL-selloff-to-ignite-gaming-expansion/2006/10/18/1160850987522.html>, last access on July 31, 2009.
- Gary Becker and Richard Posner. The Future of Newspaper, <http://www.becker-posner-blog.com/archives/2009/06/the.future.of.n.html>, last accessed on June 23, 2009.
- Ben Fritz. YouTube Loosing Less Money than Thought, Happy Hollywood Doesn't Know

- It. Los Angeles Times, June 17, 2009 - <http://latimesblogs.latimes.com/entertainment-newsbuzz/2009/06/report-youtube-losing-less-money-than-thought-and-happy-that-hollywood-doesnt-know-it.html>, last accessed on August 1, 2009.
- David Silversmith. Google Loosing Up to \$1.65 million a Day on YouTube. Internet Evolution, April 14, 2009 - http://www.internetevolution.com/author.asp?section_id=715&doc_id=175123, last accessed on August 1, 2009.
- Emma Hartley. YouTube is losing money hand over fist, says Credit Suisse. As is Twitter. Telegraph, UK, May 6, 2009 - http://blogs.telegraph.co.uk/news/emmahartley/-9721127/YouTube_is_losing_money_hand_over_fist_says_Credit_Suisse_As_is_Twitter/, last accessed on August 1, 2009.
- Miriam Steffens. Murdoch Looks for New Ways to Monetise MySpace Traffic. The Age, Australia, February 9, 2009 - <http://www.theage.com.au/news/technology/biz-tech/murdoch-looks-for-new-ways-to-monetise-myspace-traffic/2009/02/08/1234027900076.html>, last accessed on August 1, 2009.
- Larry Dignan, Sam Diaz, and Andrew Nusca. Monetising Social Media: Still an Uphill Climb. ZDNet, May 8, 2008 - <http://blogs.zdnet.com/BTL/?p=8764>, last accessed on August 1, 2009.
- Donna Bogatin. Web 2.0 Hype: Popularity Without Profits, ZDNet, November 2, 2006 - <http://blogs.zdnet.com/micro-markets/?p=620&tag=rbxccnbzd1>, last accessed on August 1, 2009.
- Mark McCrindle. Understanding Generation Y. Government of South Australia, Department of Education and Children's Services, April 2009.
- Thorsten Joachims, Laura Granka, and Bing Pan. Accurately Interpreting Click-through Data as Implicit Feedback, Proc. ACM SIGIR Int. Conf. on Research and Development in Information Retrieval, Salvador, Brazil, August 15, 2005.
- Bernard Jansen. Investigating Customer Click-through Behaviour with Integrated Sponsored and Non-Sponsored Results. Int. Journal of Marketing and Advertising, 5(1), p. 74 - 94, 2009.
- Qiankun Zhao, Tie-Yan Liu, Sourav Bhowmick, and Wei-Ying Ma. Event Detection from Evolution of Click-through Data. Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, Philadelphia, USA, August 2006.
- Meeyoung Cha, Haewoon Kwak, Pablo Rodriguez, Yong-Yeol Ahn, and Sue Moon. I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System. Proc. Int. Conf. Internet Measurements, San Diego, California, USA, October 2007.
- Jonathan Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating Collaborative Filtering Recommender Systems. ACM Transactions on Information Systems, 22(1), pp. 5 - 53, 2004.
- Jonathan Herlocker, Joseph Konstan, and John Riedl. Explaining Collaborative Filtering Recommendations. Proc. ACM Int. Conf. on Computer Supported Cooperative Work, Philadelphia, Pennsylvania, USA December 2000.
- Pedro Domingos. Mining Social Networks for Viral Marketing. IEEE Intelligent Systems, 20(1), 80-82, 2005.
- Jure Leskovec, Lada A. Adamic, Bernardo A. Huberman. The Dynamics of Viral Marketing. Proc. ACM Int. Conf. on Electronic Commerce, pp. 228 - 237, Ann Arbor, Michigan, USA, 2006.

QUEST: Discovering Insights from Survey Responses

Girish K. Palshikar, Shailesh S. Deshpande, Savita S. Bhat

Tata Research Development and Design Centre (TRDDC)

54B Hadapsar Industrial Estate Pe 411014 India.

{gk.palshikar, shailesh.deshpande, savita.bhat}@tcs.com

Abstract

Surveys are an easy, important and reliable method to measure the “pulse” of the organization’s stake-holders. A survey helps in identifying improvements to current products, services and business processes. With the advent of the Web, it is now easy to conduct large-scale on-line surveys. However, it is a challenge to analyze the responses to derive novel, interesting, actionable insights and to design effective improvement plans. In this paper, we describe a tool called QUEST for analyzing survey responses. QUEST’s pre-packaged knowledge containers provide frequently needed analysis of survey responses. Built-in analysis in QUEST varies from summaries, reports and charts to detailed statistical and data mining analysis and optimization. The analytics is designed to answer specific business questions, detect specific types of patterns, extract specific kind of useful and actionable knowledge and automatically suggest optimal improvement plans. We present a real-life case-study where QUEST was used to analyze responses from a real-life employee satisfaction survey in an IT company.

Keywords: Survey analysis, Employee satisfaction survey, data mining, text mining, Domain-driven data mining, Root cause analysis.

1 Introduction

1.1 Surveys: Motivations and Types

Knowing specific issues, problems and the likes and dislikes of employees, customers, suppliers etc. is critical for any organization. Surveys are an easy, important and reliable method to measure the “pulse” of the organization’s stake-holders. A survey helps in identifying improvements to current products, services and business processes. These improvements are directly driven from the stake-holders’ needs as reflected in the survey responses and hence are more effective in increasing satisfaction, reducing costs, improving products etc. Survey results can be used to make business decisions that are supported by findings of real issues, needs, feelings of the stake-holders.

Surveys are often designed to understand and extract specific needs of a well-defined class of stake-holders of an organization. For example, employee satisfaction

survey (ESS), product evaluation survey, customer satisfaction surveys etc.

Different media can be used to conduct surveys. In telephonic surveys, an executive contacts the target person over a telephone, asks questions and notes down the answers in a survey form. Similar activities would take place in Gallup-style person-to-person surveys. Telephonic and manual surveys impose various constraints on the survey process, limiting the effectiveness of the survey. Number of questions cannot be too many, cannot be too specific (Eg., requiring respondent to quote specific dates or figures) and so on. The number of people covered is also limited and the accuracy of responses is often doubtful, due to manual data entry. Last, but not the least, the customer is not directly responding to the questions – there is an intermediary.

With the advent of computers and the Web, these limitations are easy to remove. Many organizations are moving towards online surveys, where respondents fill up the survey questionnaire over the web or the Intranet. The responses are stored in relational databases (see Figure 1). Online surveys offer more flexibility and convenience to the respondents; they can complete the surveys at any suitable time, can save partially filled up responses etc. Finally, online surveys have a much larger reach, much lower data collection costs (Eg., avoids manual interviews or data entry) and more reliable data collection (Eg., online validations can detect simple errors in responses and prompt the respondent accordingly).

Surveys typically contain several types of questions: value, structured and unstructured. *Value questions* ask the respondent to provide specific values (Eg., dates, numbers etc.). For example, “Enter the number of years you own product X: ____”. *Structured questions* offer a few fixed options (called *domain of values*) to the respondent, who chooses one from them. For example, “Type of food you prefer: Vegetarian, Non-vegetarian”, “Rate the food quality: 1 2 3 4”. *Unstructured questions* ask the respondent to provide a free-form answer without any restrictions. For example, “Please suggest ways in which we could improve the service”.

In this paper, we treat value questions as structured questions. Questions are often grouped into *categories*; questions within a category gather responses about a specific aspect of the product, service or organization. For example, an airline customer survey might group the questions into categories like in-flight services, airport services, check-in processes, food, tickets etc. Questions in *in-flight services* category gather responses about reading material, entertainment programmes, cabin temperature, seats, cleanliness, cabin crew, announcements etc.

Some surveys ask the respondents to provide an *importance* to each question (or to each category of questions). Possible values for importance are usually fixed; Eg., unimportant, moderate, high. Answers to structured questions are often mapped (internally) to an ordered (or unordered) set of numeric values; Eg., integers 0 to N for some N . In some surveys, the respondents are allowed to not answer some questions. Lastly, questions may be classified as *negative* or *positive* and responses to these may need to be treated separately.

Some *respondent data* (i.e., information about the respondents themselves) may be collected during the survey; Eg., age, income, gender, education, location, products owned etc. Finally, some *business data* (Eg., details of products, prices, history of interactions with customers, marketing campaigns etc.) may also be available.

Once the survey responses are collected and the survey is closed (no more responses are coming in), the next task is to analyze the responses and decide the future course of action. There are basically several goals to this analysis (discussed later).

1.2 QUEST Tool for Response Analysis

In this paper, we discuss QUEST: a tool for analyzing the responses of a survey. QUEST is aimed at non-specialist users who are not experts in statistics or data mining. Hence the central idea in QUEST is to pre-package a standard knowledge container, containing frequently needed analysis of survey responses. Thus QUEST follows a *domain-driven data mining* approach and specializes the data-mining and analysis algorithms to answer specific business questions in the survey domain. The built-in analysis in QUEST varies from summaries, reports and visual charts to detailed statistical analysis and data mining analysis designed to answer specific business questions, detect specific types of patterns and to extract specific kind of useful and actionable knowledge. We have also built a companion tool (not discussed here) called SEEK that can be used to design a survey questionnaire, deploy it on the Web, collect responses etc.

One unique feature of QUEST is the integration of statistical, data mining, and text mining functionality for response analysis. Another important feature of QUEST is its focus on providing answers to commonly-encountered business questions. QUEST does not cover specialized types of surveys such as Gap Analysis Surveys, Price Sensitivity Surveys etc. These specialized surveys are done to evaluate specific business hypotheses. QUEST is oriented towards satisfaction surveys which collect feedback from employees/customers. Further, statistical and data mining analysis in QUEST currently focuses on structured responses (Eg., satisfaction levels). Next version of QUEST will provide more facilities for analysis of numeric responses (Eg., preferred price).

QUEST provides facilities for creating many different reports and charts for quickly getting high-level summary of the responses. Statistical and data mining facilities can then be used to (a) do in-depth exploration of the data (b) get specific insights and (c) analyze the responses to answer specific business questions. Text-mining facilities can be used in a similar way to summarize, group, extract and analyze textual responses. Results

from analysis of textual responses can be linked to the results from analysis done using statistical and data mining facilities. QUEST includes some optimization functionality as well.

This paper is organized as follows. Section 2 presents various types of analysis that can be performed on survey responses. Section 3 discussed the architecture and design of QUEST. Section 4 presents a real-life case-study where QUEST was used to analyze responses from a real-life ESS in an IT company. Section 5 provides some related work. We discuss conclusions and further work in Section 6.

2 Analysis of Survey Responses

2.1 Aims of Analysis

The survey responses are usually analyzed in an interactive and exploratory manner. The aims of this analysis are two-fold:

- a) get a detailed understanding of the current status of the stakeholders needs, concerns and behaviour; and
- b) design a future course of action for achieving specific improvements based on the findings in (a).

In practice, the users resort to different types of analysis to get a specific understanding and to design a specific course of action. Typical “business goals” of the analysis of an ESS are as follows (other types of surveys need similar analysis):

1. What are the major areas of concerns (or unhappiness)?
2. How does the “unhappiness” vary over employee attributes or their combinations (Eg., across age, branches, designations etc.)? Are there any unusual variations; Eg., are there any subsets of employees in specific branch, having a specific designation, who are more unhappy as compared to similar combinations?
3. Are there any inter-relationships between areas of concerns?
4. Can unhappy employees be partitioned into subgroups (or subsets), where employees within each subset share lot of common characteristics?
5. What are good predictors of employee unhappiness?
6. Identify “interesting” subsets of unhappy employees.
7. What are the root-causes of employee unhappiness?
8. Perform what-if (or impact) analysis to judge how specific changes in responses will affect overall employee satisfaction. For example, how will a 10% satisfaction increase in company transportation category among employees with designation = ITA affect the overall satisfaction level?
9. Identify optimal ways to achieve specified increase (Eg., 8%) in employee satisfaction.

Similar questions can be asked about “happy” employees. Depending on the kind of additional data available for employees much further analysis can be done. For example, if employee performance ratings and

work history are available then their relationships with satisfaction levels can be explored (Eg., do satisfied employees get better ratings? Do employees who travel abroad frequently have higher satisfaction levels?).

There are several challenges in the analysis of survey responses: large volumes, complex data and need to define specific goals for analysis. Another challenge is to design and apply appropriate techniques for specific analysis of textual responses. Combining results of analysis of structured and textual data can be difficult.

A number of simple reports and charts can be designed to summarize the basic facts regarding the survey responses. Some of these are shown in Section 4. In this section, we illustrate how statistical and data mining analysis can be performed to answer the business questions listed above for ESS. Analysis for other types of surveys is similar.

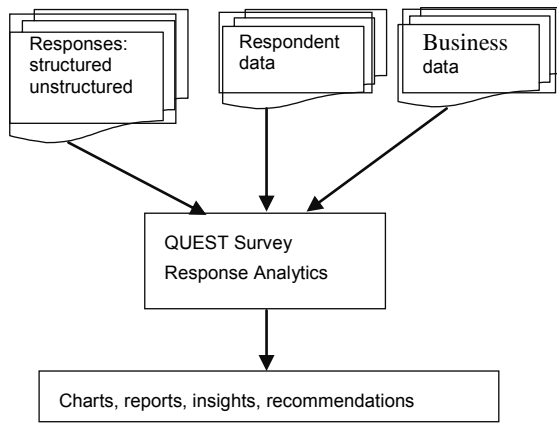


Figure 1: QUEST Survey Response Analytics Tool.

2.2 Satisfaction Index

Computing satisfaction index (SI) is important for the analysis of survey responses. Suppose a survey consists of M structured questions Q_1, Q_2, \dots, Q_M . Each question Q_j has a fixed domain D_j of possible answers; $|D_j|$ denotes the number of possible answers for Q_j . Without loss of generality, we assume that each D_j is a set consisting of numbers $0, 1, \dots, |D_j|-1$. This ordered representation of answers to a question is sometimes inappropriate for categorical questions, whose answers are unordered. For example, possible answers to the question What is your current marital status? might be unmarried, married, divorced and widowed, which cannot be easily mapped to numbers (say) $0, 1, 2, 3$. Assume that there are N respondents, each of whom has answered each of the M questions. For simplicity, we ignore the possibility that some respondents may not have answered some of the questions. Let R_{ij} denote the rating (or response) given by i th employee ($i = 1, \dots, N$) to j th question ($j = 1, \dots, M$); clearly $R_{ij} \in D_j$. Then the satisfaction index (SI) of j th question Q_j is calculated as follows (n_{jk} = no. of employees that selected answer k for Q_j):

$$S(Q_j) = 100 \times \frac{\sum_{k=0}^{|D_j|-1} k \times n_{jk}}{(|D_j|-1) \times N}$$

Clearly, $0 \leq S(Q_j) \leq 100.0$ for all questions Q_j . If all employees answer 0 to a question Q_j , then $S(Q_j) = 0\%$. If

all employees answer $|D_j| - 1$ to a question Q_j , then $S(Q_j) = 100\%$. SI for a category (i.e., a group of related questions) can be computed similarly. The overall SI is the average of the SI for each question:

$$S = \frac{\sum_{j=1}^M S(Q_j)}{M}$$

We can analogously define SI $S(i)$ for each respondent. Overall SI can be computed in several equivalent ways.

2.3 Areas of Concern

An *area of concern* is either a category (i.e., a group of related questions) or a question, having very low SI. In the simplest case, k questions (or categories) having the lowest SI can be identified as top k overall areas of concern. Areas of concerns can be also computed for specific groups of respondents (Eg., age groups, locations, designations etc.) and then compared. Following interesting insights can then be derived: (a) subsets of respondents whose areas of concerns differ from those of the entire set of respondents (b) most common areas of concerns (c) Least common areas of concerns (d) subsets of respondents that include one (or more) of the least common areas of concerns.

Let $A = \{A_1, A_2, \dots, A_P\}$ denote the set of discrete attributes of the respondents (Eg., age, gender, designation, location, experience etc.). Let V_i denote the finite non-empty set of values of attribute A_i , $1 \leq i \leq P$. Following brute force subgroup discovery algorithm for (a) systematically examines subsets of respondents and identifies those whose areas of concerns differ from those of the entire set of respondents. A randomized version of this algorithm randomly picks the subsets B of A and their descriptors Π_B . Another version of this algorithm adopts the beam search strategy to reduce the search space. Well-known Jaccard coefficient can be used to decide how dissimilar given two sets of categories are; the standard definition is adapted to take into account importance of the categories.

algorithm BF_AOC_subsets

Let C be the set of k questions having the lowest SI;

//global AOC

for $i = 1$ **to** P **do**

for each i -subset $B = \{A_{j_1}, A_{j_2}, \dots, A_{j_i}\}$ **of** A **do**

 // B is a subset of A and contains i attributes

 Let Π_B = product of the domains of attributes in B

for each i -tuple X in Π_B **do**

 Let D_X be the subset of responses satisfying X

 Let C_X be the areas of concerns for D_X

if C and C_X differ significantly **then print** X ;

endif

end for

end for

end for

For example, we might discover using the above algorithm that the subset of respondents described by $age = 30..35 \wedge designation \in \{ITA, AST\}$ has substantially different areas of concerns than the rest of the employees. We can define similar algorithms for deriving insights (b) and (c). Alternatively, to choose most common areas of

concern, we could select questions having largest support (number of respondents) in the lowest satisfaction level.

Inter-relationships (Eg., independence) between questions (i.e., areas of concerns) are often interesting. For example, sample correlation coefficient r_{ij} between questions Q_i and Q_j gives a good idea of the dependence between them. Instead of r_{ij} , we could use non-parametric statistic such as χ^2 -coefficient. Thus we can identify k pairs of questions with highest dependence between them. For example, we might find that *compensation* and *immediate supervisor* are the two most highly correlated questions (see Figure 6). This means that whenever an employee gives a low score to *compensation*, he/she is very likely to give low score to *immediate supervisor* also and vice versa. If the questionnaire contains many questions, then using r_{ij} as question similarity measure, clustering (Jain A.K., Murty M.N., and Flynn P.J. 1999) techniques (Eg., average linkage or Ward's algorithm) can be used to identify groups (clusters) of most correlated questions. Thus we might find that the questions {*compensation*, *immediate supervisor*, *appraisal*} are most highly correlated. As another type of analysis, QUEST can identify k questions most highly correlated with the SI. Such questions are *predictors* for the respondent's satisfaction level. For example, suppose that *compensation* and *appraisal* are 2 questions most highly correlated with the employee SI. Then knowing the responses of an employee to only these 2 questions, we could predict his/her final SI with good accuracy. Thus top k predictor questions are good candidates for overall areas of concerns for the set of respondents as a whole. We can also use association rule mining to find more complex dependencies among questions.

2.4 Interesting Subsets

An important goal of analysis is to find interesting subsets (subgroups) of "unhappy" respondents, such that respondents in each subgroup can be succinctly described by common (shared) characteristics. A subgroup of respondents is *interesting* if its statistical characteristics are very different from the rest of the respondents. For example, suppose a subgroup F_1 of respondents is described by $age = 30..35 \wedge designation \in \{ITA, AST\}$ and suppose also that F contains 83% unhappy respondents whereas only 34% respondents are unhappy in the set of remaining respondents. Then clearly F is interesting (see Figure 10). If such an interesting subgroup is large and coherent enough, then one can try to reduce their unhappiness by means of specially designed programmes. We have designed and used subgroup discovery algorithms (Natu M., and Palshikar G.K. 2008) for discovering interesting subgroups of respondents. We have also adapted classification techniques, such as association based classification (CBA) (Li W., Han J., and Pei J. 2001) and decision tree, for this purpose.

2.5 Predictive Models

In *supervised learning*, we are given a *training dataset* of records (Eg., employees having attributes like age, gender, designations, location, experience etc.) along with

a *class label* for each record (Eg., happy or unhappy). The well-known statistical classification problem consists of discovering classification rules which generalize the given labeled examples. These rules can then be used to predict the class label for unseen examples. Decision trees (Quinlan J.R. 1993), support vector machines (Vapnik V. 1995) and association rule based classification (Li W., Han J., and Pei J. 2001) are some of the techniques designed to discover classification rules from a labeled training dataset (Tan P.-N., Steinbach M., and Vipin Kumar 2005, Han J., and Kamber M. 2006). We discuss several ways in which statistical classification techniques can be applied to predict the satisfaction levels of respondents. First, we discretize the overall SI of each respondent to make it a class label (Eg., *unhappy*, *ok*, *happy*).

- a) **Using only responses to predict SI.** In the simplest case, we use only the response data (responses to questions) to build a predictive model for the respondent's SI. We do not use any respondent data (Eg., age, gender etc.). We might discover classification rules such as **IF** $Q_compensation \in \{0, 1\}$ **AND** $Q_immediate_supervisor = 0$ **THEN** $SI = unhappy$. Such predictive rules give a better idea of dependence between questions and overall SI.
- b) **Using respondent data and responses to predict SI.** Next, we build a predictive model for SI using both the response data (responses to questions) and respondent data (Eg., age, gender etc.). We might discover classification rules such as **IF** $Q_compensation \in \{0, 1\}$ **AND** $Q_immediate_supervisor = 0$ **AND** $designation \in \{ITA, AST\}$ **THEN** $SI = unhappy$. Such predictive rules, if they have large support and confidence, give a better idea of the concerns of various subgroups of respondents (see Figure 9).
- c) **Using only respondent data predict SI.** Lastly, we use only the respondent data (Eg., age, gender etc.) to build a predictive model for the respondent's SI. We do not use any response data (responses to questions). We might discover classification rules such as **IF** $designation \in \{ITA, AST\}$ **AND** $location = AHMD$ **THEN** $SI = unhappy$. In effect, such rules are predictive models for identifying unhappy respondents.

2.6 Root Cause Analysis

An important analysis of survey responses is concerned with identifying subsets of unhappy respondents and then identifying *root causes* for their unhappiness. Analysis (b) in Section 2.5 discovers the *surface* (or *apparent*) *causes* of unhappiness for various subsets of unhappy respondents. Textual responses are likely to contain more information about the reasons for unhappiness (see Figure 10). For example, suppose a subset of employees is unhappy about cafeteria services, which is a surface cause. Can analysis of textual answers to cafeteria related questions shed more light on *why* this subset of employees is unhappy about cafeteria? Consider textual

answers to the question *Suggest how cafeteria services can be improved*. Text clustering techniques can be used to automatically partition (group) answers to this question into *clusters*, where each cluster represents a set of related, coherent aspects. For example, the clusters may represent aspects related to cafeteria such as *more variety*, *cheaper prices*, *more cleanliness*, *extend timings*. We can now consider as if the questionnaire contains 4 more questions (one for each cluster) having answers $\{0, 1\}$ i.e., *Do you want more variety in cafeteria?* *Do you want cheaper prices in cafeteria?* etc. For example, if a respondent had wanted *more variety* and *cheaper prices*, then his responses to the corresponding two questions are 1 and 0 to the remaining two questions. Now we can repeat the predictive model of Section (2.4)(b) and get a more detailed understanding of the reasons for the respondents' unhappiness. Another possible approach is to apply (Gorsuch, R. L. 1983) to response data and identify a set of factors that explains the observed responses. Each factor would be a combination of some of the questions, which helps in postulating a specific cause for unhappiness.

2.7 More Statistical Analysis

More detailed statistical analysis can be made on the responses to draw specific inferences. Student's *t*-test can be used compare two specific groups (male vs. female; employees with experience less or more than 5 years etc.) and to decide whether SI level in one group is statistically different from that in the other group.

As another example, suppose an organization has 12 branches, with n_1, n_2, \dots, n_{12} number of employees in them. An important question is: is there any (statistically significant) difference between the satisfaction levels of these branches. That is, are the satisfaction levels in all branches similar? 1-factor ANOVA analysis of responses to ESS can be used infer whether all branches have the same average satisfaction level. If not, then at least 2 branches differ significantly in their satisfaction levels. Similar ANOVA analysis can be used about the satisfaction levels across subsets of employees. For example, 2-factor ANOVA analysis can be used to compare satisfaction levels of employees specified by (branch, designation) combinations.

2.8 What-If (Impact) Analysis

What-if (impact) analysis helps the users to judge how specific changes in responses will affect overall respondent satisfaction Eg., how will a 10% satisfaction increase in *company transportation* category among employees with *designation* = ITA affect the overall satisfaction level? Typical form of a what-if query specifies (a) a subset of respondents in terms of respondent attributes (b) a list of questions (c) change in the responses; and (d) target (currently, overall SI). An algorithm changes the responses of the given subsets of respondents for given questions as per the given change specification and computes the new overall SI.

2.9 Designing an Optimal Action Plan

An important outcome of an ESS is to design and implement an optimal action plan for improving SI of the

respondent population as a whole. Clearly, a valid action plan must contain concrete actionable suggestions that can be implemented and that are likely to lead to a substantial increase in SI. In what sense is an action plan optimal? While different optimality criteria can be set for designing such a plan, we adopt the following:

The proposed plan should lead to the desired increase in the overall average SI at the least possible "cost".

We assume that the expected increase L in the overall SI is specified by the user as input Eg., if the user wishes to increase SI from 62.0 to 65.0 then $L = 3.0$. The proposed action plan, if implemented, should achieve an increase of L in the overall SI. We assume that each category (or question) corresponds to a possible action.

SI level	No. of respondents	No. of respondents for P1 $m=10\%$
≤ 25	166	149
25 - 50	1000	917
50 - 75	6948	6353
75-100	1452	2147
SI	62.8	64.94

Table 1. SI values for a category C.

We assume that SI values for C are discretized into 4 bins (intervals): $\mathbf{B} = \{B_1 = [0, 25], B_2 = (25, 50], B_3 = (50, 75], B_4 = (75, 100]\}$. Let N denote the total no. of respondents ($N = 9566$ in the example). Let $\mathbf{b} = \{b_1, b_2, b_3, b_4\}$ denote the set of values where each b_i denotes the no. of respondents whose SI falls into bins B_i ($i = 1, 2, 3, 4$); $b_1 + b_2 + b_3 + b_4 = N$. For example, for $C = \text{Career Opportunities}$ and the data in Table 1, we have $\mathbf{b} = \{b_1 = 166, b_2 = 1000, b_3 = 6948, b_4 = 1452\}$. The grouped average $\psi_0(C) = 62.8$ is the average SI for C .

Suppose we design and implement an improvement plan for some specific category C . How do we predict the effect of such a plan on SI for C ? This is a difficult question. As a simplification, we assume the following *effect model*: $m\%$ respondents move from each lower SI level to its immediate higher SI level and no movement occurs in the "highest" SI bin. For C , if plan P1 is implemented, we expect 17 respondents (10% of 166) to move from satisfaction level 0–25 to 25–50, 100 employees move from 25 – 50 to 50 – 75 and 695 move from 50 – 75 to 75 – 100 (column 3 in Table 1). For given $0 \leq m \leq 1$ called, *upward movement fraction*, the new values for the no. of respondents in each bin are:

$$\begin{aligned} b_1' &= b_1 - mb_1; \\ b_2' &= b_2 - mb_2 + mb_1; \\ b_3' &= b_3 - mb_3 + mb_2; \\ b_4' &= b_4 + mb_3; \end{aligned}$$

Clearly, $b_1' + b_2' + b_3' + b_4' = N$. The grouped average of the new data is $\psi_m(C) = 64.94$, which is the predicted average SI for C after implementing plan P1. Thus plan P1 leads to a *benefit* $\phi(C, m) = 2.14$ ($64.94 - 62.8$) in overall SI for C .

Let $\mathbf{C} = \{C_1, C_2, \dots, C_n\}$ denote the set of n available actions (Eg., categories). Let $0 \leq M \leq 100$ be a user-specified upper limit on the upward movement percentage; Eg., $M = 50\%$ means that no more than 50% employees move from one bin to the next higher bin. Let

$\mathbf{M} = [0, M]$ denote the interval from 0 to M , inclusive. Then the set \mathbf{A} of all possible action steps is defined as $\mathbf{A} = \mathbf{C} \times \mathbf{M}$; an *action step* is a tuple (C, m) where $C \in \mathbf{C}$, $m \in [0, M]$. An **action plan** $P = \{(C_1, m_1), (C_2, m_2), \dots, (C_k, m_k)\}$ is a finite non-empty set of action steps, in which (i) an action appears at most once ($C_i \neq C_j$ for any two action steps in P) and (ii) $m > 0$ for every action step (C, m) in P ; Eg., $P = \{(\text{Career Opportunities}, 10\%), (\text{Company Image}, 20\%)\}$. Given an action plan $P = \{(C_1, m_1), (C_2, m_2), \dots, (C_k, m_k)\}$, the *total benefit* of P is $\Phi(P) = \phi(C_1, m_1) + \phi(C_2, m_2) + \dots + \phi(C_k, m_k)$.

Many different improvement plans can be proposed for C , which differ in their effect (the value of m) and in their costs. Cost not only measures the monetary spending needed to implement a plan but also refers to time, efforts, resources etc. needed. Much domain knowledge is required to construct a suitable cost function. We assume that the cost function is given as input by the user. Let \mathbf{A} denote the set of all possible action steps. A function $f: \mathbf{A} \rightarrow \mathbf{R}^+$ associates a *cost* $f(C, m)$ with any given action step $(C, m) \in \mathbf{A}$. The function f should satisfy the following properties to be a valid cost function: (i) $f(C, m) \geq 0$ for any action steps $(C, m) \in \mathbf{A}$; and (ii) f is a non-decreasing function in m for the same C i.e., $f(C, m_1) \leq f(C, m_2)$ whenever $m_1 \leq m_2$. QUEST supports a simple built-in cost function f :

$$f(C_i, m) = m * g(C_i)$$

Here $g(C_i)$ is the user-specified relative cost for category C_i , (Eg., improving canteen is cheaper than improving transport facilities). For any category, the cost increases linearly with m (larger upward movement will cost more). If $g(C_i) = 1$ for all categories, then we have a *uniform cost function* (all actions cost the same). Clearly, this cost function satisfies the conditions in the above definition of a valid cost function. More complex cost functions can be defined; but this cost function is easier for the user to specify and understand and also it simplifies the search for an optimal plan. Given an action plan $P = \{(C_1, m_1), (C_2, m_2), \dots, (C_k, m_k)\}$, the *total cost* of P is $F(P) = f(C_1, m_1) + f(C_2, m_2) + \dots + f(C_k, m_k)$.

Suppose the user's goal is to design an action plan increase the overall SI to $L\%$; Eg., if current SI is 64.82 then one possible goal would be to increase it to 70.0 (set $L = 70.0 - 64.82 = 5.18$). The problem of finding an optimal plan of action is now defined as a linear program as follows. Find a subset P of action steps such that they satisfy the constraints in the definition of an action plan and total benefit $\Phi(P) \geq L$ and total cost $F(P)$ is minimized. QUEST solves this linear program using a standard solver.

minimize $f(C_1, m_1) + f(C_2, m_2) + \dots + f(C_n, m_n)$

subject to

$m_i \geq 0$ and $m_i \leq M$ // $0 \leq m_i \leq M$, M is a constant say 50

$\Phi(P) \geq L$ // total benefit $\geq L$ (L is a constant say 5.18)

// b_i values used to compute Φ are constants

QUEST also analyzes textual responses to identify specific actionable suggestions made by the respondents for each of the categories identified in the optimal plan. For example, if the optimal plan includes an action step

of obtaining upward movement percentage of $m=20\%$ for category $C=\text{Canteen Facilities}$, then QUEST mines the textual responses and identifies all actionable suggestions related to this category (see section 2.10).

2.10 Mining of Textual Responses

In many surveys, the respondents give free-form unrestricted textual responses to some questions. Several types of analysis can be done on the responses to a specific question.

Text clustering: The responses (or sentences in them) could be grouped using text clustering techniques (Zhao Y., and Karypis G. 2005) into clusters, such that each cluster indicates a coherent type of response. For example, responses to the question Why would you recommend our company to others? could be automatically grouped into say 3 clusters – each cluster described by keywords like helpful staff, well-known brand, excellent service.

Sentiment Analysis: Sentiment analysis techniques (Pang B., and Lee L. 2008) could be used to assign a sentiment level to each response; Eg., (+, 0, -). Assuming that there are multiple questions requiring textual responses, we could aggregate (for each respondent) the sentiments of responses to individual questions into an overall sentiment. One can then test for the correlation between the overall sentiment and overall SI for the respondents i.e., is overall sentiment a good predictor of the overall SI? We could also check whether the textual responses are consistent with related questions. For example, if a respondent has strong positive sentiment in his answer to the above question, and his textual response to that question falls into the cluster labeled excellent service, then we may also expect a high rating from him for the question Rate the quality of our services: 0 1 2 3. If that is not the case, then such an exception is interesting. We could analyze such exceptions: how many exceptions each question has and how many respondents are inconsistent in their responses to text and structured questions.

Discovering important suggestions: Textual responses typically contain short and very generic sentences: staff is very helpful and friendly or he solved my problem very quickly. Such comments provide little insight into reasons for respondent's satisfaction or dissatisfaction. On the other hand, some comments are important because they are very specific and provide actionable suggestions; Eg., Please provide the facility to leave message for support executive or The fact that the correspondence stating that they would take 5000 from my bank was pretty scary. Such important suggestions can directly help in addressing specific issues and thereby in improving satisfaction. We now present text-mining techniques to automatically identify such important suggestions from given set of textual responses. We characterize each sentence using following attributes: (a) sentence length: important suggestions have substantial content compared to generic comments and hence have above average length (words). (b) Average semantic depth: important suggestions tend to use specific words; Eg., the endowment policy is not performing or the car insurance claims process is complicated. Endowment and car insurance are specific insurance products. The

semantic depth indicates if the answer has something specific to say. Semantic depth of a word is the distance of the word in WordNet (an online dictionary) concept hierarchy from the root word (such as entity). Average semantic depth of a sentence is the average of the semantic depths of all words in the sentence. More the average semantic depth, more specific the answer is likely to be. (c) Unique words: Document frequency of a word (i.e., the number of responses in which that word appears) indicates the specificity of the response in which that word appears. A word is unique if its document frequency is low (i.e., the word appears in only 1-2 responses). More the number of unique words in a sentence, more are the chances that the sentence is an important suggestion. A simple empirical rule for classifying a sentence as important or not is as follows: IF length > 10 AND $6.5 \leq \text{average semantic depth} \leq 8.0$ AND $3.27 \leq \text{count of unique words} \leq 6.56$ THEN important suggestion = 1. The algorithm to discover important suggestion computes the features for each sentence and classifies it as important or not using such rules. Some examples of discovered important suggestions are:

(a) I was phoning on my mobile and she wanted me to do a customer service survey after the call and I said no as I was paying for the call she kept going on about how it would not take long and it just irritated me so I hung up. (b) The customer service on a whole was okay but I was surprised to know that they were not prepared to tell me why my policy was not performing which made me consider cashing it in.

3 QUEST: Functionality, Architecture

QUEST is a tool for processing and analyzing responses to various types of surveys. QUEST is developed at Tata Research Development and Design Centre (TRDDC), Pune, India, which is a part of Tata Consultancy Services (TCS). QUEST is aimed at non-specialist users who are not experts in statistics or data mining. Hence the central idea in QUEST is to pre-package a standard knowledge container, which includes frequently needed types of analysis of survey responses. The built-in analysis in QUEST includes summaries, profiles, reports, various charts, KPI reports, *frequently used analytics (FUA)* designed to answer specific business questions, detection of specific types of patterns and to extract specific kind of useful and actionable knowledge. We have also built a companion tool (not discussed here) called SEEK that can be used to design a survey questionnaire, deploy it on the Web, collect responses etc.

3.1 Features

QUEST is a user-friendly application system which provides structured and unstructured data analysis capabilities.

Structured data analytics: User can easily select and work with a subset of responses (Eg., all respondents having *location* = MUM) and can create various reports for structured responses. The types of reports available with QUEST are as follows:

Satisfaction index (ASI) report: SI is an aggregated measure indicating the extent of respondents' satisfaction

(100% completely satisfied, 25% - 0% - extremely dissatisfied). As explained earlier, SI is a weighted average of the rating (responses) and importance values given by a respondent (see Figure 5).

4X4 Reports (Important categories): The notion of importance (not to be confused with importance of a category as given by the respondent) in this report is defined based on the number of respondents giving very low rating (Eg., 1 and 2 out of 4) and very high importance (Eg., 3 and 4 out of 4) to a particular category. The more number of respondents in this quadrant indicates the overall ASI is greatly influenced by these categories and hence more important the category is. The report gives sorted (descending order) list of important categories for a given intersection of data based on the above quadrant count.

Average rating and importance reports: These reports give user the sorted list of categories/questions based on their average importance or rating (see Figure 7).

Text report (for further use in clustering): The output text file generated by this functionality is used as input for text data analysis under unstructured data analysis. It is a simple report of all text comments given by an employee/customer for a particular category of open questions.

Text data analytics: QUEST also includes a text clustering functionality. QUEST can group similar respondent comments into number of clusters (groups) specified by user. Each cluster then indicates reason for satisfaction or dissatisfaction. Number of respondents in the group indicates how important a particular reason is. Instead of clustering whole responses, QUEST can build clusters out of individual sentences in responses as well. This is important because a single response may include many different types of suggestions. Then each sentence can belong to a different cluster. QUEST can also automatically identify the optimum number of clusters for the responses (see Figure 8).

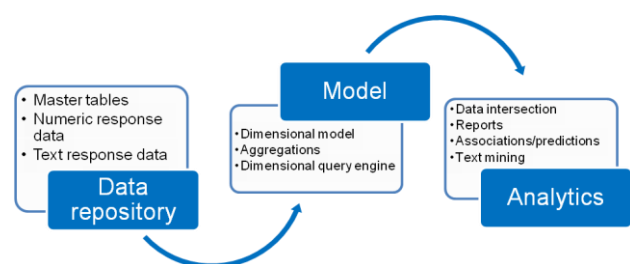


Figure 2. Architecture of the QUEST tool.

3.2 Architecture

QUEST has three main components (see Figure 2):

Data repository: The survey response data can be divided into following groups:

Employee/ Customer/ Product attribute master data - These attributes comprise of respondent attributes (age, type of responder, experience etc.), organizational structure attributes (geography, country, center etc.).

Survey attributes – The categories of questions, questions themselves, question types, their ids etc.

Answer data/Survey response collected – The actual numeric and text responses given by the respondents taking the survey

Respondent attributes are available in master tables maintained by the organization and the survey responses are available in another table. QUEST combines these two views of the data in single table and then it uses it for dimensional modeling. QUEST uses two separate tables for text and numeric data.

Model repository: QUEST uses a definition file to create dimensional model of the data and to process the aggregations. The model repository stores these model definitions and the aggregations of the numeric responses such as rating and importance given to a question. This component also provides important services for processing dimensional query and rendering the results.

Analytics function/interface: The QUEST interface is divided into two sub-menus – structured data analysis and unstructured data analysis. With structured analysis interface, user can map any dimension on rows or columns and set additional data filter and can create reports for metric like satisfaction index. The QUEST interface reads the dimension model and displays the attributes of the data for taking intersection. QUEST use MS-EXCEL component to connect to the dimensional model and aggregations. The dimensional query posed by the interface is processed and the charts are plotted by excel components.

QUEST provides text-clustering functionality through unstructured data analysis interface. QUEST uses the repeated bisection algorithm (Zhao Y., and Karypis G. 2005) to group the textual responses into various clusters. The collection of responses is converted to the standard vector space model based on TF/IDF. Repeated bisection algorithm then uses the vector space model to arrive at similar documents for forming groups.

4 QUEST: A Case Study

We present, in the following section, a case study where QUEST is been successfully used to provide insights for a real-life ESS in a large IT company (the client). The client, a large software organization, values contributions made by its associates and gives paramount importance to their satisfaction. It launches an ESS every year on its Intranet to collect the feedback for various organizational functions such as human resources, work force allocation, compensation and benefits etc. The survey contains a mixture of structured and textual questions. The goal is to analyze the responses and get insights into employee feedback which can be used to improve various organization functions. Figure 3 shows sample questions; Figure 4 shows a sample response. Figure 5 – 10 present sample results from various analysis techniques discussed earlier.

A: Leadership				
A.1 Senior management				
How important is this to you	<input type="radio"/> Extremely important	<input type="radio"/> Important	<input type="radio"/> Less important	<input type="radio"/> Not at all important
	Strongly agree	Agree	Disagree	Strongly disagree
Senior Management act as a role model	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Senior management provides clear direction for the future	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I can easily reach out to senior management through various forums	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Suggestions for improvement				
<input type="text"/>				
In My Own Words				
I like following things at my workplace				
<input type="text"/>				
I don't like following things at my workplace and suggestions to change them are:				
<input type="text"/>				

Figure 3. Sample Questions in the survey.

ID	CAT	SUBCAT	QID	IMP	RATING	SUGGESTION
AXRT	A	A.1	1	3	2	Man. is superb
AXBt	A	A.1	2	3	4	Man. is superb
AXTT	A	A.1	3	3	4	Man. is superb

Figure 4. Sample Response Table.

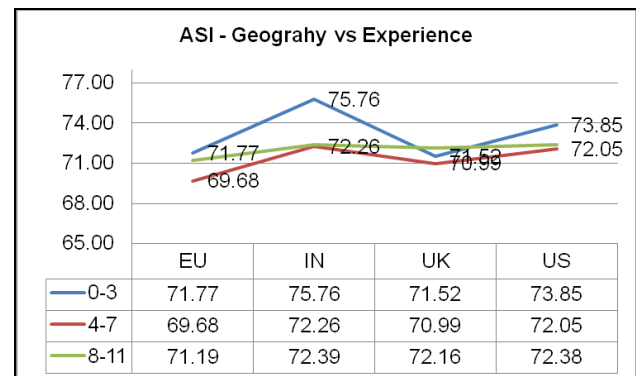


Figure 5. Satisfaction Index chart for geographies.

Correlated Items	Description
23 And 7	23:- Organizational Communication, 7:- Customer and Market Focus
24 And 5	24:- Performance Appraisal/ Management, 5:- Compensation and Benefits
23 And 8	23:- Organizational Communication, 8:- Digitization and Process Improvements
17 And 7	17:- Job Content, 7:- Customer and Market Focus
18 And 23	18:- Learning and Development, 23:- Organizational Communication

Figure 6. Dependency between Categories

Category	Average Importance
Immediate Supervisor	3.68
Compensation and Benefits	3.63
Payroll	3.63

Figure 7. Important Categories (average importance).

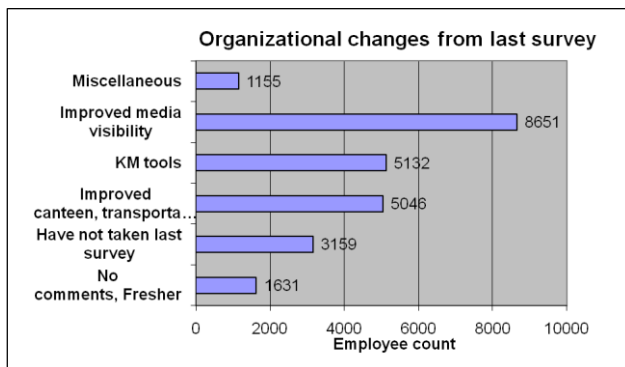


Figure 8. Clusters of textual responses to organizational changes.

Prediction rule: **IF** EXPERIENCE RANGE = '1-3' **AND** GEOGRAPHY = INDIA **AND** GENDER = Male **THEN** RATING for C5 = 1

Figure 9. Model to predict rating.

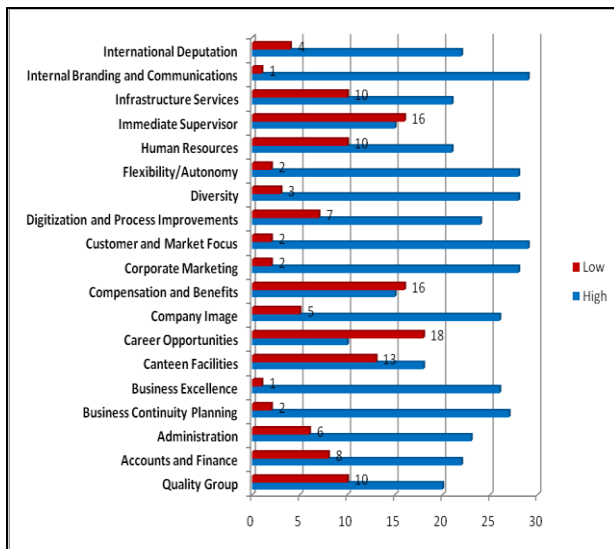


Figure 10. Root-cause Analysis.

QUEST was used to apply the association-rules based classification algorithm (CBA) to survey responses. This algorithm discovered the following association rule (among many others) which describes a subset of 29 unhappy employees.

customer=X AND designation=ASC => ASI < 65 (13/29)

This rule states that employees having designation=ASC and who are working at Customer X are unhappy. The chart in Fig. 10 then tries to find the root causes of the unhappiness of that particular subset of employees. Basically, this chart identifies the categories which are rated very low by these employees. As seen, the employees in this subgroup are significantly unhappy about categories career opportunities and compensation and benefits. Further insights into the root causes can be obtained by analysis of their responses to relevant questions.

5 Conclusions and Further Work

With the advent of the Web, it is now easy to conduct large-scale on-line surveys. However, analyzing the responses to derive novel, interesting and actionable

insights to design effective improvement plans remains a challenge. In this paper, we described a tool called QUEST for analyzing survey responses. QUEST's pre-packaged knowledge containers provide frequently needed analysis of survey responses. Built-in analysis in QUEST varies from summaries, reports and charts to detailed statistical and data mining analysis designed to answer specific business questions, detect specific types of patterns and to extract specific kind of useful and actionable knowledge. We presented a real-life case-study where QUEST was used to analyze responses from a real-life ESS in a large IT company.

We are working on enhancing the built-in analytics in QUEST and on providing a better alignment of the analytics with the business goals of conducting and analyzing the survey. In particular, we are interested in adding facilities to allow the users to state different types of (statistical) hypotheses regarding the responses, which the tool can then verify or reject. We are also working on automatically building statistical models of subsets of respondents. Incorporating more data mining techniques (Eg., anomaly detection, clustering and sequence mining) is also of interest. Another significant area of further work is better integration of text and data analytics. We also wish to provide a framework to allow the user to easily build specialized "recipes" containing sequences of analytics. Finally, we wish to link the results obtained from analysis of survey responses with applications such as customer churn prediction.

6 References

- Gorsuch, R. L. (1983): *Factor Analysis*. Hillsdale, NJ, Lawrence Erlbaum.
- Han J., Kamber M. (2006): *Data Mining: Concepts and Techniques 2/e*. Morgan Kaufmann.
- Jain A.K., Murty M.N., Flynn P.J. (1999): *Data Clustering: a Review*. ACM Computing Surveys, 31(3): 264-323.
- Li W., Han J., Pei J. (2001): CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules. in *Proc. 2001 IEEE Int. Conf. on Data Mining*, Nov. 29-Dec. 02, 369-376.
- Natu M., Palshikar G.K. (2008): Discovering Interesting Subsets using Statistical Analysis. in *Proc. Int. Conf. on Management of Data (COMAD2008)*, Dec. 17-19, 2008, Mumbai, India, 2008.
- Pang B., Lee L. (2008): Foundations and Trends in Information Retrieval. In *Opinion Mining and Sentiment Analysis*. 2(1-2), 1-135.
- Quinlan J.R. (1993): *C4.5: Programs for Machine Learning*. Morgan Kaufmann.
- Tan P.-N., Steinbach M., Vipin Kumar (2005): *Introduction to Data Mining*. Addison-Wesley.
- Vapnik V., (1995): *The Nature of Statistical Learning Theory*. Springer-Verlag.
- Zhao Y., Karypis G. (2005): Hierarchical Clustering Algorithms for Document Datasets. *Data Mining and Knowledge Discovery*. 10(2): 141 - 168.

Discovering inappropriate billings with local density based outlier detection method

Yin Shan, D. Wayne Murray, Alison Sutinen

Program Review Division, Medicare Australia
134 Reed St. North, Tuggeranong
ACT 2900, Australia

{Yin.Shan, Wayne.Murray, Alison.Sutinen}@medicareaustralia.gov.au

Abstract

This paper presents an application of a local density based outlier detection method in compliance in the context of public health service management. Public health systems have consumed a significant portion of many governments' expenditure. Thus, it is important to ensure the money is spent appropriately. In this research, we studied the potentials of applying an outlier detection method to medical specialist groups to discover inappropriate billings. The results were validated by specialist compliance history and direct domain expert evaluation. It shows that the local density based outlier detection method significantly outperforms basic benchmarking method and is at least comparable, in term of performance, to a domain knowledge based method. The results suggest that the density based outlier detection method is an effective method of identifying inappropriate billing patterns and therefore is a valuable tool in monitoring medical practitioner billing compliance in the provision of health services.

Keywords: local outlier factor, LOF, health data mining, fraud detection, open source data mining.

1 Introduction

In many countries, public health systems have consumed a more significant portion of governments' expenditure than ever and this trend is likely to continue as the aging population will unavoidably require more medical resources. Thus, there is an urgent need to make sure that the scarce funding available for the public health service is spent appropriately. This usually involves analysing a large amount of data collected by the public health agencies. Data mining can provide one of the major instruments for exploring health service data. (Becker,

Kessler and McClellan, 2005, Lin *et. al.*, 2008, Yang and Hwang, 2006).

One of main concerns of public health agencies is the occurrence of inappropriate practice and fraud. Given the large amount of spending on public health, a small percentage of non-compliance would result in a huge loss of the public money. The US Department of Health and Human Services, estimated that improper Medicare benefit payments made during 2002 financial year totalled \$13.3 billion, or about 6.3 percent of the \$212.7 billion in processed fee-for-service payments reported by the Centers for Medicare and Medicaid Services (CMS) (US DHHS, 2003). Therefore it is critical to detect these non-compliant activities to facilitate the proper and efficient use of the resources.

In Australia, a government agency, Medicare Australia, administers Medicare, a fee for service national health funding system for Australians. There are over 400 million transactions processed by Medicare Australia and approximately 30 billion dollars benefit paid in 2007-2008 (MA 2008). Medicare Australia is also responsible for undertaking reviews to ensure the integrity of associated health programs it administers. There have been a series of studies that have applied a range of data mining techniques to the Medicare Australia data for various compliance purposes (Pearson, Murray and Mettenmeyer 2005, He, Graco and Yao 1999, Shan *et. al.*, 2008). In this research, we focus on detecting inappropriate billing of one medical speciality group.

In this compliance domain, it often occurs that there is a lack of labelled cases, which makes it difficult to employ supervised machine learning techniques, while outlier detection becomes attractive. This work demonstrates that a local density based outlier detection method (Breunig *et.al.*, 2000) significantly outperforms a few other methods and can be effectively used in the detection of inappropriate practice.

The remaining sections of this paper are organised as follows. The problem domain is briefly introduced in Section 2. Section 3 provides a brief background on outlier detection and density based method. The data set used is described in Section 4, whilst Section 5 outlines the experimental studies applied to the data set. The results and its evaluations numerically and by subject

Copyright (c) 2009, Australian Computer Society, Inc. This paper appeared at the Eighth Australasian Data Mining Conference (AusDM 2009), Melbourne, Australia. Conferences in Research and Practice in Information Technology, Vol. 101. Paul J. Kennedy, Kok-Leong Ong, and Peter Christen, Eds. Reproduction for academic, not-for profit purposes permitted provided this text is included.

matter experts are covered in Section 6. Section 7 presents discussions and the last section is the conclusions and future research.

2 Optometrists Billing Compliance

Optometrists claim a small yet significant portion of Medicare benefits. There are approximately 3,000 Optometrist practicing in Australia. In contrast to the group of General Practitioners (GPs) which has over 25,000 practitioners nation-wide, this is a relatively small group in size.

This group is quite unique in terms of the number of Medicare items it can bill. There are only 26 items in the Medicare Benefit Schedule (DHA, 2007) which optometrist can bill whilst there are commonly hundreds of items available to other groups of medical professionals.

Usually, the proportion of these items should be relative stable and we understand there are a small number of factors which may contribute to the variations, such as optometrist's speciality (e.g. some optometrists specialised in contact lens fitting) and patient age. These items bear different level of compliance risk. There is a possibility that some items are overused resulting in over servicing or some items are inappropriately used to obtain more financial benefit to the optometrist. This research hypothesizes that by detecting outliers in terms of item usage and a few other factors, the individuals with unusual billing practice may be discovered. These individuals may bear higher risk of inappropriate practice as they are significantly different from their peers.

3 Local density based outlier detection method

The concept of outlier detection originally came from the field of statistics (Hawkins, 1980). These methods are usually very well understood and have solid theoretical grounds. However, they often require the prior knowledge of probability distribution and more importantly they are usually univariate. In the domain of data mining we more often than not encounter huge datasets with at least dozens of variables, whose underlining distributions are unknown.

Recently there have been a number of algorithms proposed to address the above issues of the traditional statistical methods. Some of them are based on clustering methods (Ester, et al, 1996, Ng and Han 1994, Zhang et, al 1996). Outliers in those clustering methods are defined as those records, which cannot fit neatly into any clusters and thus need to be identified and treated as exceptions. Some other methods have been specifically designed to detect these outliers (Knorr and Ng 1998, Ramaswamy et al , 2000, Breunig et al 2000).

In this work, a local density based outlier detection method (Breunig *et.al.*, 2000) is used. In this method, one single measure, the Local Outlier Factor (LOF), indicating the degree of outlier-ness is calculated for each record. The records having the largest LOF values are the most significant outliers. There are several reasons to choose this method. Firstly, this method is based on the local property, which is suitable for this problem. A series of K-means clustering analyses were performed on this data set with different number of clusters and there was no clear global cluster structure found, as evident by very small value of the silhouette information (Rousseeuw, 1987). The silhouette information measures how well the points are grouped into clusters. If the value is close to 1, it suggests the points in general clearly belong to certain clusters. The value is close to -1, otherwise. Small silhouette values obtained in our experiments undermines the basis of employing most of the clustering based outlier detection methods, as there is no obvious cluster discovered in this dataset. Secondly, this method has only one parameter to tune and requires minimum prior knowledge, such as probability distribution, which is unknown. Thirdly, this method provides one straightforward rating of the degree of outlier-ness - LOF.

The complete formal definition of LOF can be found in the original paper (Breunig *et.al.*, 2000). An accessible and simplified description is presented here for the completeness of the paper.

We assume that for any object p , there are no two objects that are the same distance from that object p . This greatly simplifies the discussion without compromising the basic idea. We need several simple notions before introducing the definition of Local Outlier Factor (LOF). For any positive integer k , the k -distance of object p , denoted as $k\text{-distance}(p)$, is the distance of k -th object from p . The k -distance neighbourhood of p , denoted as $N_{k(p)}$ contains every object whose distance from p is not greater than the k -distance. The local reachability density of p , denoted as $lrd_k(p)$, is simply the inverse of k -distance(p):

$$lrd_k(p) = \frac{1}{k\text{-distance}(p)}$$

The local outlier factor is just the average of ratios of local reachability density of p to that of its neighbouring objects:

$$LOF_{MinPts}(p) = \frac{\sum_{o \in N_{MinPts}(p)} \frac{lrd_{MinPts}(o)}{lrd_{MinPts}(p)}}{MinPts}$$

where $MinPts$ is a predefined constant, which can be loosely interpreted as a smoothing factor. It is easy to see that the lower p 's local reachability density is, and the higher the local reachability densities of p 's $MinPts$ -nearest neighbours are, the higher is the LOF value of p . From this definition, it is clear that LOF is basically a

measure comparing the density of region around the object we are interested to the densities of those regions of its surrounding objects. This definition implies that outlier-ness is a local property and it is very intuitively appealing when a global cluster pattern does not exist or is not the focus. For more justification and comparison, please refer to the original publication (Breunig *et.al.*, 2000).

There are many distance measures, such as Hamming distance and Euclidean distance. This local density based method is not limited to any particular measure. We choose Euclidean distance in this research.

4 Optometrist data set

The data set used in this study was drawn from Medicare Australia's Enterprise Data Warehouse, covering billing records of optometrists for a rolling one year period, ending the third quarter of 2008 (1 Oct, 2008 – 30 Sept, 2008 inclusive).

There are only 26 items in the Medicare Benefit Schedule (Australian Government, 2007) which optometrists can bill Medicare Australia. Given the advice from the domain expert, some of these 26 items are combined thus we obtain a list of 12 unique variable combinations from the original 26. Each variable represents the number of services of a particular item or combined items. The patient age is obviously related to some optometry services and thus average patient age is included as an extra variable. The total number service is also included, resulting in 14 variables for each record. The data set contains 2893 optometrists, one record for each optometrist. The density based outlier detection and some other comparing methods are performed on this dataset.

5 Experimental study

We applied two major analyses on this data set, the density based LOF outlier detection method and a domain knowledge based univariate method, to discover inappropriate practice. In addition to these two methods, a random sampling was also conducted to serve as a baseline measure, as any more complicated method has to outperform this baseline.

1) LOF method. There is no prior knowledge of relative importance among the variables and thus we assume that each of these 14 variables has equal weight. The data is normalised before feeding into the LOF algorithm. We used one of common normalisation methods, i.e., each variable is normalised to mean 0 and standard deviation 1 by subtracting its mean and then dividing by its standard deviation (Sarle, 2002). There is only one tuning parameter *MinPts* in the LOF algorithm as explained in Section 3. We choose a lower bound of *MinPts* 30 and an upper bound of 50. We followed the heuristics offered in (Breunig *et.al.*, 2000) to determine the LOF, i.e., the data is fed into the LOF algorithm with two *MinPts* values, 30

and 50, the resulting larger LOF value is taken as the final value for that record. The LOF calculation was undertaken using the *dprep* package in the open source statistical software R.

2) Domain knowledge based univariate method. This is a crude univariate method. Based on the experience of the domain expert, amongst the 14 variables in the dataset, there are three variables which may be particularly related to high risk behaviour. If any of these three variables significantly deviates from its mean, it is often an indication of higher risk. So in this method, we pick those optometrists who have values for any of these three variables that are 4 or more standard deviations away from its mean. This gives us a list of 32 optometrists.

3) Baseline random sampling. A sample of 25 optometrists was drawn randomly as the most basic benchmark.

We propose two approaches to compare the results and assess the effectiveness of the methods in identifying potential non-compliant individuals in this research. The first validation method is indirect. The compliance history of identified optometrist is analysed. The intuition is that there should be some correlation between high risk optometrists and past records of non-compliance activities. The second one is a direct method. The identified optometrists were presented in a de-identified form to the subject matter expert for evaluation.

Admittedly, none of these two validation methods are perfect as the only relatively reliable method of validating non-compliant individuals is comprehensive desk and field audit, which is often prohibitively costly if it needs to be done on a large scale. The first method - validation using historical compliance records - has the implicit assumption that the past is an indication of the future which is obviously not always true. The second method requires the involvement of the subject matter expert, which takes advantages of prior human knowledge and at the same time presents the opportunity for human error. Furthermore, it is possible that a particular subject matter expert might tend to focus on a particular set of compliance risks. However, we speculate that the combination of these two complementary methods would give us an indication of the performance of the outlier detection methods.

6 Results

The results from three methods – density based LOF, domain knowledge based univariate method and baseline random sampling are compared with two approaches – history checking and domain expert manual validation. These two comparisons are presented in the following two subsections respectively, which give us similar conclusions.

6.1 Validation using historical compliance records

There is a compliance database available to us, containing records of medical practitioners and allied health

professionals who have been approached in relation to previous compliance activities. The first validation is to match specialists identified by three methods against their compliance history in this compliance database. This provides an estimate of the effectiveness of the various outlier detection methods studied in this work in detecting non-compliant practice.

Method	Have compliance record
LOF - High	36.00 %
LOF - Low	0.00 %
Univariate	31.25 %
Random	25.00 %

Table 1: Comparison of three methods. LOF method is presented in first two rows.

In Table 1, the results of the three methods are compared – density based LOF method, univariate method and random sampling. As discussed previously, Local Outlier Factor (LOF) is a measure of outlier-ness. Large LOF value indicates large deviation. However, is large deviation directly correlated to a high risk of potential compliant activities? In order to verify this, 25 records with highest LOF values and 25 records with the lowest LOF values were examined, as listed in the first row of Table 1. It is evident that the LOF value is a good risk indicator, as 36% of records with highest LOF have compliance record while none of the records with lowest LOF have a compliance record. It is clear from Table 1 LOF is significantly better than randomly sampling and at least as good as crude univariate method.

Method	Average number of compliance records per optometrist who has at least one compliance record
LOF - High	1.67±1.12
LOF - Low	N/A
Univariate	1.30±0.67
Random	1.00±0.00

Table 2: Comparison of three methods on the average number of compliance records per optometrist who has at least one compliance record.

We are aware that an optometrist may have different numbers of records in their compliance history, which suggests some of them have multiple incidents of non-compliant practice or have been engaged in multiple compliance activities. We speculate that the average number of compliance records per optometrist may be an indication of the severity or certainty of a possible non-compliant practice. So we listed the average number of compliance records per optometrist who has at least one compliance record for each method in Table 2. For all those optometrists identified by the highest LOF values

and who had compliance records, they had on average 1.67 records (with the standard deviation 1.12). While this is higher than 1.30 of the univariate and 1.00 of the random sampling, the difference is not statistically significant.

We also experimented larger sample size. We reduced the threshold of the univariate method to 3 standard deviations and that resulted 109 records and we drew same number of records with the highest LOF values. The average number of compliances records per optometrist from univariate and LOF methods was still different but not statistically significant.

Checking against an optometrist's compliance history provides us with a measure to compare the density based LOF method with other methods. In this comparison, we find that LOF is at least as effective as the univariate method in identifying high risk individuals and significantly outperforms random sampling. However, if the certainty of a possible non-compliant practice can be measured by the number of compliance records one has, there is no significant correlation between LOF value and the number of compliance record per optometrist.

6.2 Validation by domain expert

Checking compliance history of an optometrist is an indirect way of validation. During this study we had access to a domain expert, a compliance optometrist, to provide us with a more direct evaluation. Although the best way to validate whether an individual optometrist was truly non-compliant, would be a review by a panel of peers consulting a domain expert was the best we had access to at the time.

We invited the domain expert to evaluate each group of de-identified optometrists and rate each individual with one of three levels of risks – low, medium and high. The evaluation is listed in the Table 3.

Method	High	Medium	Low
LOF - High	6	16	3
LOF - Low	0	0	25
Univariate	13	18	1
Random	0	8	17

Table 3: Rating of optometrists provided by subject matter expert.

As listed in Table 3, the majority of optometrists identified by LOF (with highest value) and univariate methods are rated as having a risk level of medium or above, whilst the randomly sampled optometrists were mostly rated as posing a low risk.

To gain a better idea of these results, we regroup these three groups of risk levels into two. Medium and high risks are grouped together, and classified as *Unexplainable*, as the behaviour of these individuals cannot be explained by the domain expert and thus may

be related inappropriate practice activities. Low risk optometrists were reclassified as *No Further Action* because they look normal and no further action needs to be taken.

The outcome of this regrouping is listed in Table 4. It clearly shows that density based LOF is a good indication of risk as 88% of individuals with highest LOF are risky and none of the individual with low LOF represent a risk. Compared to other methods, the LOF method significantly outperforms random sampling, which is consistent with the results from the previous history checking validation.

Method	Unexplainable	No Further Action
LOF - High	88.00%	12.00%
LOF - Low	0.00%	100.00%
Univariate	96.887%	3.13%
Random	32.00%	68.00%

Table 4: Risks of optometrists provided by domain expert.

From the Table 4, we can see over 97% of optometrists identified by univariate method cannot be explained while only 88% by LOF in this evaluation. However, this needs to be interpreted with care. As mentioned before, the univariate method is derived from the experience of the domain expert and is based upon unusual values for one or more of the three variables believed to be related to high risk behaviour. When the domain expert evaluates the risk, these are the most obvious indicators to look at. Not surprisingly, the univariate method has an accuracy approaching 100% in this evaluation. On the other hand, the density based LOF method considers multiple variables at the same time and thus the results are not as straightforward for the domain expert to interpret. With these considerations in mind, we argue that LOF might have comparable performance as univariate methods in the real world if the bias is removed in this analysis. Furthermore, LOF method looks at multiple factors simultaneously. It is possible that it might help identify individuals with multiple compliance issues or subtle issues that univariate method cannot identify.

7 Discussion

Combining the results from two evaluation methods – compliance history checking and domain expert validation, we can see that LOF is significantly better than the baseline random sampling and has the comparable accuracy of the univariate method. However a number of questions remain, including, does similar performance suggest that the LOF and the univariate method identify similar groups of people? Is LOF just an alternative way to employ a domain knowledge based univariate method of discovering individuals with extreme values of single variable? There are many different types of non-compliant practice and thus we do not expect one single method would be able identify all of these different types. Therefore, in this research, what may be useful is to investigate whether there is a

significant overlap between high risk individual identified by LOF method and the domain knowledge based univariate method. If there is a significant overlap, that would render the LOF redundant.

We matched the list of 25 individuals identified by LOF with highest LOF values against 32 individuals identified by the univariate method with extreme values. There is only one individual appearing in both and thus the overlap is minimum, which indicates LOF is not a simple replication of univariate analysis on this problem.

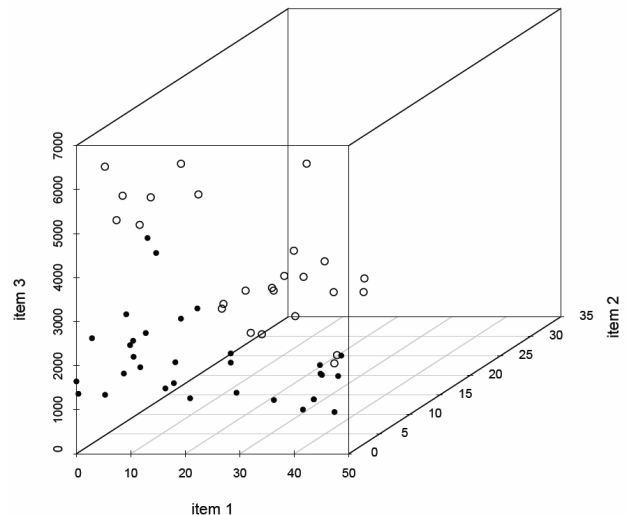


Figure 1. High risk individuals identified by LOF and the univariate method. Circles are from the univariate and solid dots are from LOF.

To further demonstrate that the LOF and univariate analysis actually identify different types of high risk individuals. The individuals identified by LOF and the univariate method are plotted against three variables which are used by univariate method in Figure 1. It is clear that those individuals identified by different methods cover different regions of the problem space, which shows that these two methods are complementary to each other.

8 Conclusions and future research

This paper presents a novel application of density-based local outliers and demonstrates how it can be used with the aims of detecting inappropriate practice in the health service management domain, in this case, using optometrist billing compliance.

The results suggest the density based LOF outlier detection method is effective. We validated the results in two major ways. The first validation method is matching against compliance history and the second is the domain expert confirmation. Although there are different emphasis and bias for both of validation methods, they provide us similar conclusions.

These validation methods shows that LOF is a reliable indication of high risk individuals i.e. where low LOF values clearly related to individuals behaving consistent

to their peers and high LOF values clearly related to higher risk. The LOF method significantly outperformed the baseline method, which was randomly sampling of optometrists. It is also at least as effective as a univariate method derive from prior domain knowledge. Furthermore, LOF method looks at multiple factors at the same time. It is possible that it might help identify individuals with multiple compliance issues or subtle issues that univariate method cannot identify.

In practice, as one of the outlier detection methods, the density based method can be integrated into a bigger system, which employs multiple methods to detect inappropriate practice, for better accuracy. From our experience, the hybrid approach involving both automatic data mining techniques, such as LOF, and domain experts' theory driven methods, often facilitate us constructing very effective systems for compliance purposes. Thus, the future work includes further analysis on the effectiveness of LOF methods and, if appropriate, how to best integrate it with other methods in Medicare's operational systems.

9 Acknowledgements

The authors would like to acknowledge the input of Dr. David Jeacocke for his insightful comments during this research, and Dr Steve Zantos for his advice on the interpretation of optometric data.

10 References

- Becker, D. and Kessler, D. and McClellan, M. (2005) Detecting Medicare abuse. *Journal of Health Economics*. **24**(1): 189-210.
- Breunig, M.M., Kriegel, H-P, Ng, R.T. and Sander, J. (2000) LOF: Identifying Density-Based Local Outliers. pp 93 -104 in *Proceedings of the ACM SIGMOD 2000 International Conference on Management of Data*, Dallas, Texas
- Department of Health and Ageing. Medicare Benefit Schedule Book. (2007) ISBN 1-74186-363-5, Department of Health and Ageing, Australian Government. Canberra. ISBN 1-74186-363-5
- Ester M., Kriegel H.-P., Sander J., Xu X. (1996) A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, pp. 226-231 in *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining*, Portland, Oregon, AAAI Press, .
- Hawkins, D. (1980) Identification of outliers. Chapman and Hall, London.
- He H., Graco, W. and Yao X. (1998) Application of Genetic Algorithm and k-Nearest Neighbour Method in Medical Fraud Detection. pp. 74-81 in *Second Asia-Pacific Conference on Simulated Evolution and Learning (SEAL '98)*, Canberra, Australia., LNAI 1585 Springer.
- Knorr, E.M. and Ng, R.T. (1998) Algorithms for Mining Distance-Based Outliers in Large Datasets. pp. 392-403 in *Proceedings of the 24th International Conference on Very Large Data Bases*, New York, USA, Morgan Kaufmann Publishers, San Francisco, CA.
- Lin, C., Lin, C-M., Li, S-T. and Kuo, S-C. (2008) Intelligent physician segmentation and management based on KDD approach. (2008) *Expert Systems with Applications*. **34**(3): 1963—1973. Pergamon Press, Inc. Tarrytown, NY, USA
- Medicare Australia. (2008) Medicare Australia Annual Report 2007-2008. ISSN 0313-1041
- Ng R. T., Han J. (1994) Efficient and Effective Clustering Methods for Spatial Data Mining. pp. 144-155 in *Proceedings of the 20th International Conferences on Very Large Data Bases*, Santiago, Chile, Morgan Kaufmann Publishers, San Francisco, CA.
- Pearson, R., Murray, W. and Mettenmeyer, T. (2006) Finding Anomalies in Medicare. *Electronic Journal of Health Informatics*. **1**(1): e2. (www.ejh.net/ojs/index.php/ejhi/issue/view/1)
- Ramaswamy, S, Rastogi, R. and Shim, K. (2000) Efficient algorithms for mining outliers from large data sets. pp 427–104 in *Proceedings of the ACM SIGMOD 2000 International Conference on Management of Data*, Dallas, Texas
- Rodríguez, C. (2004) A computational environment for data preprocessing in supervised classification. M.Sc. Thesis, University of Puerto Rico, Matagüez. Puerto Rico.
- Rousseeuw, P.J. (1987) Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, **20**: 53–65.
- Sarle, W. T. FAQ of Usenet newsgroup comp.ai.neural-nets 2002. [Online] Available <ftp://ftp.sas.com/pub/neural/FAQ2.html> September, 2009
- Shan, Y., Jeacocke, D., Murray, D.W. and Sutinen, A. (2008) Mining medical specialist billing patterns for health service management. pp 105 – 110 in Roddick, J.F., Li, J., Christen, P. and Kennedy, P. (eds) *Conferences in Research and Practice in Information Technology*. **87**.
- US Department of Health and Human Service. (2003) Improper Fiscal Year 2002 Medicare Fee-for-Service Payments. A-17-02-02202
- Yang, W-S. and Hwang, S-Y. (2006) A process-mining framework for the detection of healthcare fraud and abuse. *Expert Systems with Applications*. **31**: 56–68. Elsevier
- Zhang T., Ramakrishnan R., Linvy M. (1996) BIRCH: An Efficient Data Clustering Method for Very Large Databases. pp.103-114 in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, ACM Press, New York.

Predictive analytics that takes in account network relations: A case study of research data of a contemporary university

Ekta Nankani¹Simeon Simoff^{1,2}

¹ School of Computing & Mathematics
University of Western Sydney,
Email: enankani@scm.uws.edu.au

² Email: s.simoff@uws.edu.au

Abstract

Contemporary organisations incorporate large amount of invisible networks between their employees. The structure of such networks impacts the information fusion within the organisation. Taking into account the influence of such network structures in predictive modeling will be beneficial for the quality of organisational strategic planning. Network mining methods (the social network analysis of large heterogeneous data sets) can extract information about the structure of such networks and the strategic positioning of each individual from various interaction data. We propose to integrate the output of network mining into the predictive modeling cycle in order to depict these influences. This paper demonstrates such approach by incorporating network centrality measures of actor closeness and actor betweenness in CART predictive modeling cycle. It presents a proof-of-concept application of this integrated approach to the case study of a contemporary university, which resembles some similarity with corporate organisations. The study utilises a data set about academic research activities collected over five years. The results of the study support the hypothesis that information about the network structures in a data set (whose impact is included through the centrality measures) can improve the accuracy of predictive analysis.

Keywords: Predictive Analytics, Social Network Analysis (SNA), Centrality Measures, Data Enrichment

1 Introduction

The research direction taken in this work has been inspired by the visionary research by Tapscott and Williams (Tapscott & Williams 2006) on challenging the deeply rooted assumptions about the role of competitiveness and collaboration in business and society as a whole. "The four principles – openness, peering, sharing and acting globally – increasingly define how twenty-first-century corporations compete." ((Tapscott & Williams 2006), p.30). Their new economic vision draws a picture of a world of a business collaboration on a massive scale as a key to survive in a globally competitive environment. Remaining innovative requires understanding the shifts in the environment and the development of new strategies that foster collaboration in order to progress in a com-

petitive environment. This reality faces both industry and academia. Technology advances are based on the advance of fundamental sciences. Contemporary research and development activities in industry are tighten to the need of being fast, efficient and capable of earning clear return on investment. However, innovations continue to rely on fundamental knowledge, hence, industry will increasingly rely on partnerships with universities and other research organisations, leaving corporate research teams to move quickly to technology development and practical application. In practice, close cooperation between industry and academia potentially can enable the industry partners to keep their edge, while spreading the upfront research and development costs across a much broader ecosystem (see (Tennenhouse 2004) for an example of the implementation of such strategies).

An essential strategy in making the most out of such partnerships is the deepening and broadening collaboration across research communities, starting with fostering strategic collaborations within a university. This is the practical problem that motivates the research in predictive modeling presented in this paper.

Understanding the structure of existing and predicting potentially new collaborations is vital when it comes to enabling the interaction between industry and academia. This interaction as well the interaction and combination of several disciplines, are seen as the key drivers of contemporary innovation. Consequently, critical becomes the development of robust business intelligence methods that can

- extract essential information and knowledge about the structure of collaboration;
- produce reliable models that can be used for prediction (recommendation) of new collaborative ventures.

In depicting collaboration these analytics methods and respective technologies have to deal with heterogeneous data about academic activities that link academics into various invisible networks.

In this paper we have focused on predictive modeling that contributes information to the processes that support the development of research directions in universities. The paper presents early proof-of-concept results in two aspects of academic activity:

1. obtaining internal research funding, and;
2. type of research output in terms of publication categories.

The first one looks at predicting whether a research project proposal, submitted to one of the university research grant schemas will be funded or not. The second one looks at predicting the most-likely DEST category of publications in which academic

Copyright ©2009, Australian Computer Society, Inc. This paper appeared at the Eighth Australasian Data Mining Conference (AusDM 2009), Melbourne, Australia. Conferences in Research and Practice in Information Technology, Vol.101. Paul J. Kennedy, Kok-Leong Ong, and Peter Christen, Eds. Reproduction for academic, not-for profit purposes permitted provided this text is included.

publications will fall into, e.g whether these publications will be books, book chapters, journal articles or conference papers.

Both of these tasks involve researchers from the same organisation, hence, the assumption is that *the structure of collaborative relations between researchers matters*. Consequently, the paper explores two ways of predictive modeling for each of the above tasks:

- conventional predictive modeling without taking the structure of collaborative relations;
- extended predictive modeling, which takes in account information about the structure of relations.

The presentation in the paper is centred around this practical problem in order to demonstrate the practical value of proposed solutions. Further the paper is organised as follows: Section 2 looks at two modeling perspectives - predictive analytics and social network analysis, in the context of the problem. Section 3 considers some of network centrality measures as carriers of information about the positioning of and relations between network structural elements. Section 4 uses a dataset about research activities of academics in an Australian university to present the integrated approach and methodology for addressing the above formulated predictive tasks; Section 4.5 discusses the results of the analysis; and Section 5 considers the future developments and concludes the paper.

2 Modeling perspectives in analytics

Contemporary knowledge economies and digital ecosystems rely on capturing and utilising diverse data about the activities and processes within them. Consequently, a continuously growing variety of data analytics techniques addresses the need for converting these data into useful information for decision making purposes. This section provides a brief overview of the two modeling perspectives in analytics that are relevant to presented work: predictive modeling (predictive analytics) and social network analysis for competitive intelligence.

2.1 Modeling perspective in predictive analytics

In data analysis models which are used to predict future data trends are known as predictive analysis models. Classification or estimation algorithms are central in predictive analytics and are used in many areas of human endeavour, including (but not limited to) business and science. Examples of application areas from business include credit approval, medical diagnosis, performance prediction and selective marketing. Predictive models assess unlabeled samples to determine the value or value ranges of an attribute that a sample is likely to have (Han & Kamber 2001). With predictive analysis the validity of the classification result (and the true accuracy of the model) can be verified by waiting for the future event to happen. Though predictive accuracy is a critical aspect of models there are other facets that are equally important. We may require a model to show which of the predictor variables are most important in the dataset (Smyth 2001). We may be interested in examining whether predictor variables interact or whether a simple model can result in good prediction. In the research, presented in this paper, we are interested in taking in account the structure of "social" relationships between the entities in a predictive modeling

dataset. In particular, we consider enriching the predictive modeling dataset with attributes that represent information about the structure of such relationships. Such attributes are based on concepts from social network analysis (SNA). In this paper we append attributes that correspond to some SNA centrality measures and then test the hypothesis that *appending centrality measures improves the prediction accuracy*. For the purpose of the paper, the dataset we use is a snapshot of a five year span, that, to some extent, encapsulates the temporal relationship of predictors to the target variable (Linoff 2004).

Any of the classification or estimation techniques can be used for predictive analysis on the proposed enriched dataset. The five criteria for evaluating predictive methods include predictive accuracy, computational speed, robustness, scalability and interpretability (Han & Kamber 2001). Proposed enrichment of the dataset affects four of these criteria. On the positive side is the expected improvement of predictive accuracy and interpretability of the results. However, the approach requires additional computation of centrality measures, which will affect the computational speed and scalability.

We show in Figure 1 an example of a fragment of data about academics and research students in a university, similar to the one considered in the case study, to position the approach presented in the paper. The data set includes the following attributes: Name (of the person), Position (in the university), School (as administrative unit), Research Center (for those involved in research centers), Publication type (according to DEST classification), Co-Authorship (indicates a list of people from the same set that have published together with the person in consideration), Co-Supervision (indicates a list of people from the same set that have co-supervised higher degree research students).

'Conventional' predictive modeling deals with the portion of the data set, contoured by the double dotted line. Let the task be the prediction of the type of publication in which most off the output of a researcher will be falling into. Hence, the attribute "Publication type" is selected as the "output" ("target") attribute and the attributes "Position", "School", "Research Center" are the "input" ("predictors"). As the aim is to derive a general trend, the attributes that contain unique identifiers, such as "Name", will not be taken in account. As a result, 'conventional' predictive modeling cycle does not have mechanisms to take into account some of the relations that may exist between the instances in the data set - in our case, between the individuals. The issues and problems of depicting such dependencies with predictive analytics methods have been discussed in the context of network mining in (Simoff & Galloway 2008). The chapter considered two groups of issues:

1. the "loss of detail" - the hidden links existing between the instances in a data set;
2. the assumption about the independency of the attributes of a data set.

In this paper we deal with networks that are explicitly encoded. For instance, a co-authorship relation between two researchers can be define as the association of the names as authors on the same paper. Though it may not accurately and in-depth reflect the actual authorship in terms of contribution and development of the research work, it reflects the underlying assumption that co-authorship involves some interaction and information exchange between the authors. In terms of data analysis this means some embedded dependency between the instances that represent

these researchers in the data set. Centrality measures are one way to represent explicitly this dependency. Next section presents aspects of social network analysis relevant to the approach presented in the paper.

2.2 Modeling perspective in social network analysis

Social networks represent groups of people, various connections among them and the dynamics of such connections for in-depth analysis (McDonald April 2003). The production of knowledge is a social process involving interactions among people and organisations with different backgrounds, resources, predispositions and insights (von Krogh et al. 2001, Tushman & Rosenkopf 1992). Measuring these heterogeneous social networks is done to study the influence of emergent social structures within and external to an organisation on the business and engineering processes within it.

Social network analysis and network mining are means that address the problem of discovering organisational intelligence from existing and potential interactions in the organisational settings. Traditional social network analysis usually deals with networks where only “cognitive agents” (people, groups of people, the human capital of organisations) can be the nodes. Network mining can be viewed as an extension of SNA, not just in terms of the volume of the data, but also in terms of the content of the network models: nodes can be any elements, including resources, expertise, intellectual property, technologies, products, markets).

According to Wasserman and Faust, “Social Network Analysis (SNA) provides a formal conceptual means for thinking about social world” (Wasserman & Faust 1994). Contemporary SNA deals with the analysis of interactions between social entities in an organisation, based on large data sets of human interactions (Shetty & Adibi 2005). SNA research recognises the elements of an organisation as intentional networks, hidden networks, socially translucent networks, mediators, and structural holes. These elements can depict changes with processes in a group over a period of time. Important for our work is that through such elements and the respective model parameters in network analysis methods we obtain information about the structural inter-dependence in an organisation, that is, “who knows who”, and “who knows what” (Srivastava et al. 2006), which to some extent reveals structures of information fusion. This comes from the fact that our daily life is very much influenced by social networks through which we interact with various groups of people: family, friends, colleagues. Through these networks we indirectly connect to people associated with these groups without necessarily knowing them. The need to take into account these interactions has been recognised in several areas of applied modeling in information systems, including viral marketing, e-mail filtering based on social networks, various recommender systems (Matsuo et al. 2007).

The study of social networks formed on social networking sites, such as orkut, flickr, youtube, myspace, can help to detect the most influential users. Many properties of the social network have been studied: Pool and Kochen scientifically formulated the small world phenomena (Schnettler 2009); according to Milgram the average path between two Americans is six hops (Schnettler 2009); Granovetter suggest that social networks can be partitioned into strong and weak ties, with strong ties tightly clustered (Granovetter 1973); nodes with high indegree also tend to have high outdegree, showing active members are also popular members (Mislove et al. 2007).

Study of social influence is a strategic arena for SNA research. Some argue that influence is a special instance of causality, namely the variations of one person's responses by the actions of another (Stanley Wasserman 1994). SNA approach and techniques are not limited to humans and can be used to study a variety of phenomena (Wasserman & Faust 1994), hence the increased interest in the academic community (Kumar et al. 2006). Contemporary SNA is associated mostly with visual analysis of graph structures (McDonald April 2003).

As mentioned in Section 2.1 our method brings the SNA modeling perspective into predictive modeling. It considers the estimated underlying graph models from a portion of a data set that usually would be ignored in predictive modeling cycle. Several parameters of such models are included in the extended data set for predictive modeling. The practical grounds for taking such approach are motivated from previous network studies which indicate that the social structural contexts surrounding actors shape a variety of responses both attitudinal and behavioural. In customer analytics, for example, behavioural features are believed to be more reliable than demographics. Behavioural targeting is to target right person at right time, hence the drive for developing methods that can produce more accurate predictions of customer behaviour. Logically such methods should utilise information from various networks in which customers can be involved, including alumni, referrals, rehires and business development (Drakos et al. 2008). In an organisation the inferred networks assist in identifying the knowledge flow and find out solutions for corporate related problems, sometimes even the extent to which an individual has succeeded in performing his work (Heer 2004).

In this paper we consider the utilisation of information about collaborative networks, which are important drivers of the knowledge flows within organisations (Singh 2005), including universities and research institutions. Scientific networks are an example of collaborative networks that has a long history of investigation, in particular, citation networks have been studied as knowledge flow structures in sciento- and bibliometrics. More recently, the focus has shifted to co-authorship networks in order to get a better understanding of the underlying structure of knowledge evolution. Relevant to the underlying philosophy of our work is the use of a regression method to estimate the probability of knowledge flow between inventors of any two patents (Hu & Jaffeb 2003).

Our work is also inspired by “the law of the few” or the “80/20 principle” (Gladwell 2000). According to Gladwell, “the success of any kind of social epidemic is heavily dependent on the involvement of people with a particular and rare set of social gifts.” In other words in any situation roughly 80 percent of the ‘work’ will be done by 20 percent of the participants. Gladwell divides these ‘20 percent’ into three types:

- *Connectors* - those ones that “bringing the world together” as a result of their ability to span many different worlds;
- *Mavens* - those ones that connect people with new information, i.e. the information brokers;
- *Salesmen* - those ones that persuade people.

The difference between these types of actors in social networks is reflected in their positions and patterns of linking. Hence, the inclusion of social network measures in the training dataset enables the capture of such information in the predictive model. In the next section we discuss some of the centrality measures, that have been utilised in this study. Rather

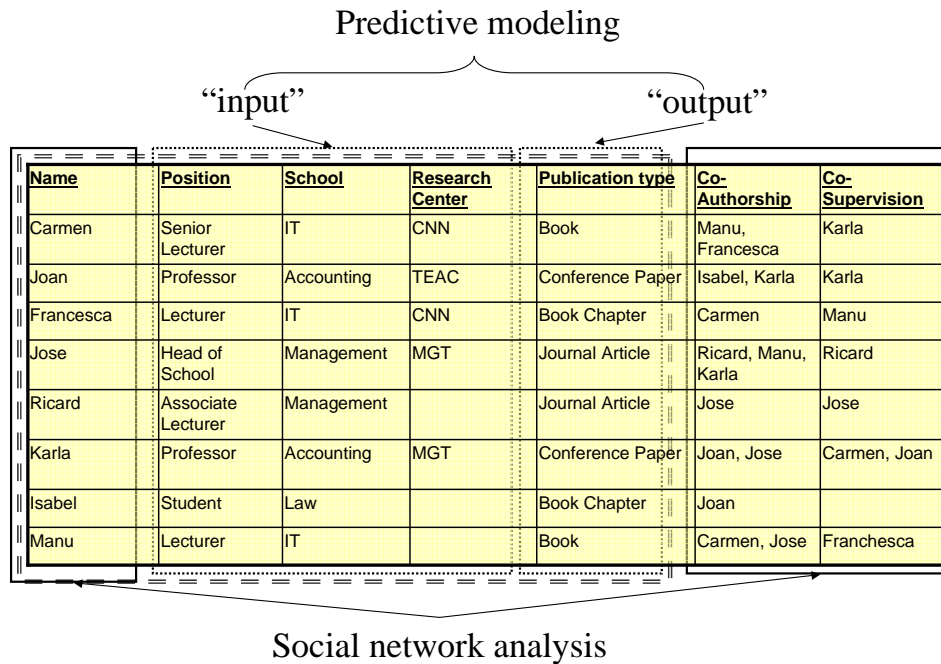


Figure 1: “Predictive modeling” and “social network analysis” perspectives of a data set

than from a graph-theoretical point of view, we discuss the role of these measures in the network models from a domain perspective.

3 Information about network structural elements reflecting relations

When dealing with publications in broad sense, including not only papers but also postings on various usenet groups and blog sites, the analysis of network relations is as essential as the text analysis of the content. For instance, in (Agrawal et al. 2003) researchers demonstrated that link analysis can be more valuable than text-based algorithms when it comes to classification of people on two sides of an issue in a usenet group.

Analysis of centrality measures determines the importance of vertices in a network based on their connectivity within the network structures. For instance, in health science centrality measures help researchers in depicting the structure of underlying biological networks that model biological processes as complex systems; the approach has been successfully applied to different biological networks (Dwyer et al. 2006).

Social network relations are measured within a set of actors. In this paper we consider a single mode network - a network described by a dataset that contains information about only one type of actors - in this case these are people. The actors include academic staff, researchers, students and externals, associated with university. The relationship between actors is a kind of professional collaboration, and includes *co-authorship*, *co-supervision*, *co-teaching*, and *co-participation* in a project.

There are different measures to quantify network relationships. These measures help to test propositions about network properties rather than simply relying on descriptive statements. To understand the role of an actor in a network SNA evaluates the location of actors (nodes) through a set of centrality measures. These measures provide information about the different aspects of actors' role in a network with respect to their position, e.g. connectors, bridges, leaders, isolates, as well as about the clusters in the network structure and which actors are in them, which

actors form the core of the network, and which actors reside on the the periphery.

Centrality of an actor is measured in terms of actor *degree*, *closeness* and *betweenness*. *Actor degree* refers to the number of links an actor has. The idea behind actor degree is that actors with more links are in a more independent position - such actors are less dependent on any specific actor. In terms of collaborative research networks, high values of actor degree measures may indicate more administrative research role (e.g. a research director) than a research collaborator role in terms of ideas flow, hence actor degree measures are not taken in account in the current work.

Actor closeness measures the ability of an actor to reach other actors in a network at shorter path lengths, or, reciprocally, actors who are more reachable by other actors at shorter path lengths. In terms of collaborative research networks this structural advantage can be translated into potential for initiating research collaboration, e.g. starting a project or initiating co-supervisory arrangements.

Actor betweenness measures the ability of an actor to broker contacts among other actors in the network, e.g. the extent to which an actor is positioned between the other actors. In terms of collaborative research networks this structural advantage can be translated into potential for growing research collaboration, e.g. extending an existing team of chief investigators for the next grant application, amalgamating research groups into a larger entity.

In this study we consider four centrality measures:

- three closeness measures: closeness, eigenvector centrality and harmonic closeness, and;
- betweenness.

The brief description of these measures is presented below following (Wasserman & Faust 1994).

Closeness measured as the length of the shortest-path, scores higher values to more central vertices. Closeness at actor n_i level is calculated as

$$C_C(n_i) = \left[\sum_{j=1}^g d(n_i, n_j) \right]^{-1}$$

where, $C_C(n_i)$ is *Closeness* of n_i and $d(n_i, n_j)$ is the number of links in the geodesic path linking actors i and j (that is $d(node1, node2)$ is a distance function) and this sum is from $j = 1$ to $j = g$, where g refers to all the other actors not including i actor. This index is the inverse of the sum of the distances from actor i to all other actors. In terms of information flow, those actors with highest closeness values are well positioned for monitoring the information flow in the network. In a collaborative research network research leaders are expected to be in such positions.

Eigenvector centrality (known also as eigenvector of geodesic distances) is another form of closeness, looking for the most central actors in terms of the overall structure of the network. From a factor analysis perspective, the eigenvector centrality measure ranks actors in terms of some new dimensions that characterise the distances among actors, where the first of this new dimensions captures the positioning of an actor with respect to the overall network structure, and the rest are depicting more local sub-structures. An eigenvalue in this context defines the location of each actor with respect to each dimension, hence, the term eigenvector when considered with respect to all actors in the network. The measure of centrality is computed as the largest positive eigenvalue. The eigenvector centrality measure for n_i is

$$C_{EV}(n_i) = \sum (C_{EVmax} - C(n_j)) / C_{EVmax}$$

where, $C_{EV}(n_i)$ is eigen vector for n_i , C_{EVmax} is max eigen vector and this sum is for all the actors from $i = 1$ to $i = j$.

Harmonic closeness is an alternative measure of closeness that takes in account all the pathways that connect an actor to all others, rather than just the geodesic. The measure estimation is based on an algorithm that uses the harmonic mean length of paths ending at the given node. In a collaborative research networks broad collaborators are expected to be in positions with high value of this measure.

Betweenness depicts those actors that occur on many shortest paths between other actors, having higher betweenness than those that do not. Similar to the other centrality measures, there is a family of betweenness measures - the one used in our study is

$$C_B(n_i) = \sum_{j < k} g_{jk}(n_i) / g_{jk}$$

where, $C_B(n_i)$ is Betweenness of n_i and g_{jk} is geodesic linking two actors i and j . The actor betweenness centrality for n_i is sum of estimated probabilities over all pairs of actors not including i th actor. Actors with high betweenness can be power players, but can be also the single point of failure. In a collaborative research network, for instance, their removal may cause fragmenting (up to disintegration) of the network.

In an earlier work (Nankani, Simoff, Denize & Young 2009) we have focused on the discovery and analysis of network structures in university data about academic activities. The method relied on a combination of network mining techniques with substantial visual analysis and qualitative data analysis for validation purposes. The work has analysed networks at different levels of granularity, varying from individual level through to networks between divisions. In this paper we use only the network structures of relations between individuals. In the next

section we use a case study format to test the hypothesis that information about the network structures in a data set (whose impact is included through the centrality measures) can improve the accuracy of predictive analysis.

4 Case study of university research data set

The case study in this section is based on an integrated university research data. It demonstrates the integrated approach of social network mining combined with predictive analysis on two predictive modeling tasks:

1. *forecast internal research grant application outcome* - whether a research project will get funding or not, and;
2. *predict the predominant category of personal publication output* - whether an academic will be publishing predominantly conference papers, journal articles, book chapters, books or any other category of creative work registered in the data set.

The completion of both tasks depends on numerous factors beyond the scope of the dataset. By incorporating the centrality measures we are looking at developing a feasible approximation for performing these predictions.

4.1 Description of the data set

Table 1 shows the description of the data set, which includes integrated data about a range of different academic activities, including

- co-authorship;
- co-participation in a research project;
- co-supervision of research students;
- other related academic data, which is not taken in account in this work.

All data are time-stamped, collected over a consecutive span of 5 years. During this period of time the university in consideration has had 9 schools and 23 research centres. All collaborative ties are between staff, students and external participants.

Readers can find more details about some of the results of the network analysis in (Nankani, Simoff, Young & Denize 2009) [these include details about the evolution of the networks over a time span and analysis of centrality measures, with network visualisations generated with NetDraw graph visualisation tool].

4.2 Methodology

We divided this project into three different stages, as shown in Figure 2

The purpose of each of these phases is detailed as follows.

Phase 1 includes integrated data collection, cleaning, developing an understanding of the data structures and composing the original data set for the analytics tasks. Details of Phase 1 are discussed in (Young et al. 2008, Nankani, Simoff, Denize & Young 2009, Nankani, Simoff, Young & Denize 2009).

Phase 2 includes

Data Statistics

Description	Number of Records
Data Records collected	24,556
Clean Data Records	15,177
Number of Distinct Nodes	2,131
Number of Ties	37,398

Table 1: Description of the university research data set

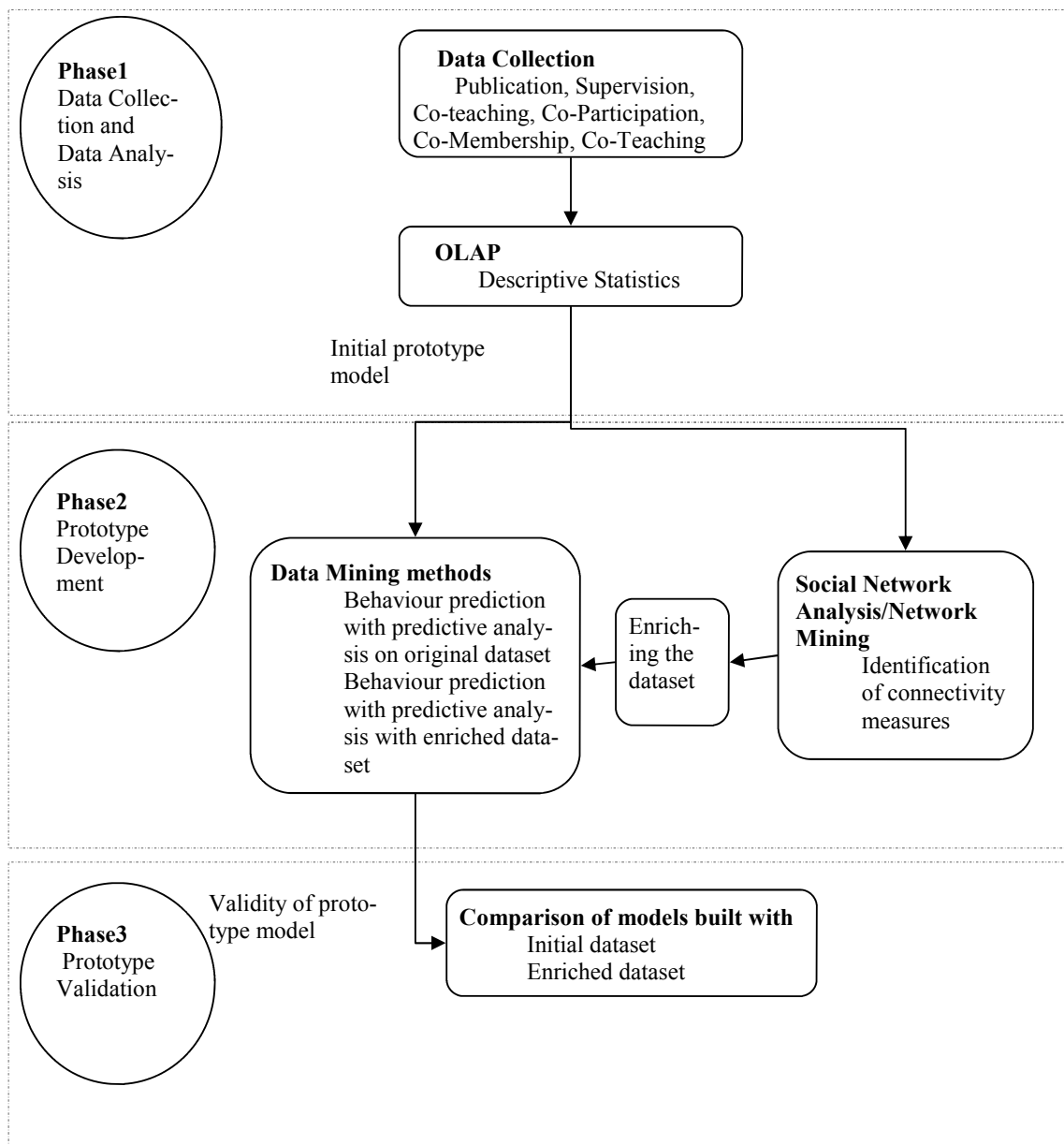


Figure 2: Methodology

1. creating predictive model based on the the original data set;
2. performing social network analysis based on the original dataset (work presented in in (Young et al. 2008, Nankani, Simoff, Denize & Young 2009, Nankani, Simoff, Young & Denize 2009);
3. enhancing the original data set by appending centrality measures;
4. creating predictive models based on the enhanced dataset.

Phase 3 includes the analysis and comparison of the models, created with the original and enhanced data sets.

4.3 Generating centrality measures

The centrality network measures of closeness, eigenvector, harmonic closeness and betweenness are estimated with the respective algorithm implementations in UCINET social network analysis software. These network measures then are appended as additional attributes to the existing academic data set, extending the dataset with data about the network structures.

4.4 Predictive modeling

For this study we looked at type of classifiers that have relatively poor predictive power, but are good in handling mixed types of data and missing values, and are insensitive to monotone transformation of inputs and robust to outliers in the input space. Last but not least - classifiers with good level of interpretability of the results. Based on these criteria we have selected tree classifiers as in general they have poor predictive power and meet the rest of the criteria (see (Hastie et al. 2001), Table 10.1). CART classification tool by Salford Systems(Salford_Systems n.d.) is well suited for the purpose of the study, as it implements classification and regression trees (for some details see (Linoff 2004)). Figure 3 illustrates the steps taken to create the predictive models that address tasks 1 and 2 discussed in the beginning of section 4.

4.4.1 Predicting project funding (Task 1)

These models involved 13 attributes from the original data set, including **Person Name**; **Person Code**; **Type** (with the following possible values: 'internal member' (from the same university), 'external member' (from other university or industry) and 'student'); **Faculty** (to which the person belong to), **School** (to which the person belong to); **Year** (when a project started or a publication was made); **Research center membership**; **Project name**; **Project Status** (whether funding grant application has been approved or rejected); **Publication category**.

Model for predicting project funding based on the original data set

Predictive models were created with a sample from the original data set (sample of records are taken because of restriction of software used). All the project data was divided into 4 files with 800 records in each data set. Several random sets of 800 records with exclusion were taken at a time and respective predictive models were created and tested. The model created with the following attributes to predict Project Status whether project will be approved (funded) or rejected (not funded)

Variable Importance

Importance	Original dataset	Enhanced dataset
1	Person Name	ProjectName
2	Project Name	Research Center
3	School Name	Local Eigenvector
4	Faculty Name	Closeness
5	Year	Harmonic Closeness
6	Research Center	Betweenness

Table 2: Variable Importance - Project Funding

scored the best result. The attributes used with the model include **Research center membership**; **Person Name**, **School Name**, **Project Name**; **Year** and **Faculty Name**.

Predictive Accuracy

Learn Dataset - 81.46%

Test Data Set Using

Cross Validation - 54.62%

Separate test file - 55.77%

Model for predicting project funding based on the enhanced data set

Another set of predictive analysis models are created with enhanced data set (sample of records are taken because of software restriction). A random 800 records were taken and several predictive models were created. The model created with the following attributes to predict **Project Status** whether project will be approved (funded) or rejected (not funded) scored the best result. The attributes used with the model are **Project Name**, **Closeness**, **Betweenness**, **Eigen Vector**, **Harmonic closeness**, **Local Eigenvector**, **Year**, **Research center membership**.

Predictive Accuracy

Learn Dataset - 64.83%

Test Data Set Using

Cross Validation - 60.32%

Test data - 60.72%

Variable importance for predicting project funded or not funded

Table 2 displays all the variables in order of importance, with most important as 1 to least important as 6, for the analytical models created with the help of project data set.

4.4.2 Predicting publication category (Task 2)

This task started with the same data set as for Task 1.

Model for predicting publication category code based on the original data set

Predictive analysis models for publication category code are created with the publication data set

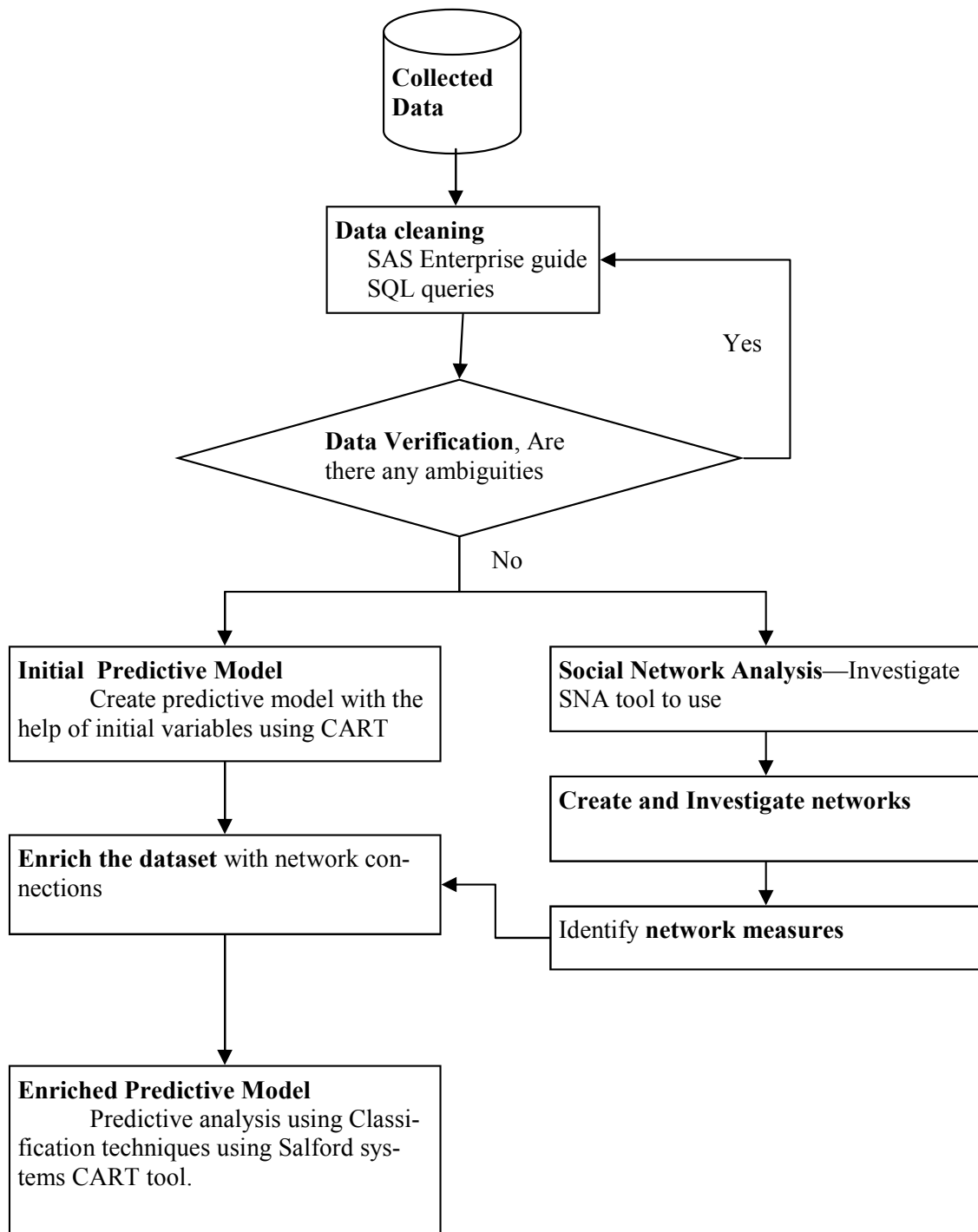


Figure 3: Predictive Modeling Steps

Variable Importance

Importance	Original dataset	Enhanced dataset
1	Person Name	Closeness
2	School Name	Harmonic Closeness
3	Faculty Name	Faculty
4	Year	Betweenness
5	Research Center	Year
6	Type	Eigenvector

Table 3: Variable Importance - Publication Category

from the original data and a sample of records are taken (sample of records are taken because of software restriction). A random 800 records were taken in a dataset and several predictive models were created and tested. All publication dataset is divided into 10 data files. Models are tested in several iterations to include all records for model testing purpose. The model created with the following attributes to predict publication category code scored the best result. The attributes used with the model include **Research center membership; Person Name, School Name, Year, Faculty Name, Type** (Person type).

*Predictive Accuracy**Learn Dataset - 83.69%**Test Data Set Using**Cross Validation - 68.42%**Separate test file - 69.72%****Model for predicting publication category code based on the enhanced data set***

Another predictive analysis models are created with enhanced data set (sample of records are taken because of software restriction). Again a random 800 records were taken and several predictive models were created and tested. The model created with the following attributes to predict publication category scored the best result. The attributes used with the model include **Closeness, Betweenness, Eigenvector, Harmonic closeness, Local Eigenvector, Faculty Name**

*Predictive Accuracy**Learn Dataset - 78.6%**Test Data Set Using**Cross Validation - 76.53%**Test data - 74.10%****Variable importance for predicting publication category code***

Table 3 displays all the variables in order of importance, with most important as 1 to least important as 6, for the analytical models created with the help of project data set.

4.5 Comparison of the results

A summary of the results from the analytical models in this pilot study are presented in Table 4. The results from the experiments with the original and extended data sets in Task 1 are presented in columns *Task 1.0* and *Task 1.e*. The results from the experiments with the original and extended data sets in Task 2 are presented in columns *Task 2.0* and

Task 2.e. These results illustrate that when the estimated SNA centrality measures of one part of the data set are added as complementary predictors to the other part of the data set they improve the prediction accuracy both in a cross validation setting and on unseen data.

Therefore, this preliminary study supports the hypothesis that information about the network structures in a data set can improve the accuracy of predictive analysis. To some extent this approach can be viewed as enhanced predictive analytics technique.

5 Conclusions

Since the days of the six-degree separation experiment, social network analysis has advanced significantly, thanks to the prevalence of online social websites and their capabilities of collecting data about the communities created around them, as well as the availability of a variety of offline large-scale social network systems such as collaboration networks. There are several technologies to support rich social interactions as blogs, wikis, social bookmarks, social tagging and these techniques are finding their way into business environments (Drakos et al. 2008).

This paper addressed the issue of utilising the information about the network structures of relations between the instances of a data set in predictive modeling cycle. Such practical problems emerge in various corporate settings, as well as in academic collaboration in universities.

The work presented a method that deploys SNA methods for extracting the structure of the network. Essential information about this structure is encoded through the various network centrality measures. In this work we have depicted four measures.

The results of this study support the hypothesis that information about the network structures in a data set (whose impact is included through the centrality measures) can improve the accuracy of predictive analysis. In both predictive tasks we have improved the average percentage accuracy over the test data. Though the improvement in the accuracy is several percent, the additional data preprocessing for estimating respective centrality measures is worth considering in domains like cancer treatment, where every percent of increased accuracy matters!

References

- Agrawal, R., Rajagopalan, S., Srikant, R. & Xu, Y. (2003), Mining newsgroups using networks arising from social behavior, in 'WWW '03: Proceedings of the 12th international conference on World Wide Web', ACM, New York, NY, USA, pp. 529–535.
- Drakos, N., Mann, J., Cain, M. W., Andrews, W., Knox, R. E., Valdes, R., Rozwell, C., Bradley, A., Maoz, M., Otter, T., Harris, K., McGuire, M., Bell, T., Basso, M., Prentice, B., Smith, D. M., Fenn, J., Prentice, S., Sarnier, A., Dunne, M. & Harris, M. (2008), Hype cycle for social software, Research ID Number: G00158239, Gartner. The Social Software Hype Cycle highlights the most important technologies that support rich social interactions. Use our assessment of their business relevance and maturity to guide your investment decisions.
- Dwyer, T., Hong, S.-H., Koschützki, D., Schreiber, F. & Xu, K. (2006), Visual analysis of network centralities, in 'APVis '06: Proceedings of the 2006 Asia-Pacific Symposium on Information Visualisation', Australian Computer Society, Inc., Darlinghurst, Australia, Australia, pp. 189–197.

Datasets used for prediction	Predictive accuracy (%)			
	Task 1.o	Task 1.e	Task 2.o	Task 2.e
Learn dataset	81.46	64.83	83.69	78.6
Test dataset using cross validation	54.62	60.32	68.42	76.53
Test dataset using test data	55.77	60.72	69.72	74.10

Table 4: Summary of the results

- Gladwell, M. (2000), *The tipping point: How little things can make a big difference*, Little Brown.
- Granovetter, M. S. (1973), 'The strength of weak ties', *The American Journal of Sociology* **78**(6), 1360–1380.
- Han, J. & Kamber, M. (2001), *Data Mining: Concepts and Techniques*, Morgan Kaufmann.
- Hastie, T., Tibshirani, R. & Friedman, J. (2001), *The elements of statistical learning*, Springer.
- Heer, J. . (2004), Exploring enron: Visualizing anlp results, in 'University of California, Berkely'.
- Hu, A. G. Z. & Jaffeb, A. B. (2003), 'Patent citations and international knowledge flow: the cases of korea and taiwan', *International Journal of Industrial Organization* **21**(6), 849–880.
- Kumar, R., Novak, J. & Tomkins, A. (2006), Structure and evolution of online social networks, in 'KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, New York, NY, USA, pp. 611–617.
- Linoff, M. J. A. B. G. (2004), *Data Mining Techniques: for marketing, sales, and customer relationship management*, Wiley Publishing.
- Matsuo, Y., Mori, J., Hamasaki, M., Nishimura, T., Takeda, H., Hasida, K. & Ishizuka, M. (2007), 'Polyphonet: An advanced social network extraction system from the web', *Web Semant.* **5**(4), 262–278.
- McDonald, D. (April 2003), 'Recommending collaboration with social networks: A comparative evaluation', *ACM* **5**, 5–10.
- Mislove, A., Marcon, M., Gummadi, K. P., Druschel, P. & Bhattacharjee, B. (2007), Measurement and analysis of online social networks, in 'IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement', ACM, New York, NY, USA, pp. 29–42.
- Nankani, E., Simoff, S., Denize, S. & Young, L. (2009), Enterprise university as a digital ecosystem: Visual analysis of academic collaboration, in 'DEST2009'.
- Nankani, E., Simoff, S., Young, L. & Denize, S. (2009), *Information Systems: Modeling, Development, and Integration*, Springer Berlin Heidelberg, chapter Supporting Strategic Decision Making in an Enterprise University Through Detecting Patterns of Academic Collaboration, pp. 496–507.
- Salford_Systems (n.d.), Cart a robust decision tree tool for data mining, predictive modelling, data preprocessing, Technical report, Salford Systems.
URL: www.salfordsystems.com/doc/CARTtrifold.pdf
- Schnettler, S. (2009), 'A structured overview of 50 years of small-world research', *Social Networks* **31**(3), 165–178.
- Shetty, J. & Adibi, J. (2005), Discovering important nodes through graph entropy the case of enron email database, in 'ACM, KDD 2005'. Chicago.
- Simoff, S. J. & Galloway, J. (2008), Visual discovery of network patterns of interaction between attributes, in S. J. Simoff, M. H. Boehlen & A. Mazeika, eds, 'Visual Data Mining: Theory, Techniques and Tools for Visual Analytics', Vol. 4404 of *LNCS*, Springer Verlag, Heidelberg., pp. 172–195.
- Singh, J. (2005), 'Collaborative networks as determinants of knowledge diffusion patterns', *Management Science* **51**(5), 756–770.
URL: <http://ssrn.com/paper=628281>
- Smyth, D. H. . H. M. . P. (2001), *Principles of Data Mining*, MIT.
- Srivastava, J., Pathak, N., Mane, S. & Contractor, N. S. (2006), Knowledge perception analysis in a social network, in 'Workshop on link analysis counter terrorism and security 22nd Apr 2006, Bethesda, Maryland.'
- Stanley Wasserman, J. G., ed. (1994), *Advances in Social Network Analysis: Research in the social and behavioral sciences*, SAGE.
- Tapscott, D. & Williams, A. D. (2006), *Wikinomics: How Mass Collaboration Changes Everything*, Portfolio, Penguin Group.
- Tennenhouse, D. (2004), 'Intel's open collaboration model industry-university partnerships', *Research Technology Management* **47**(4).
- Tushman, M. & Rosenkopf, L. (1992), *Organizational Determinants of Technological Change: Toward a sociology of Technological Evolution*, Vol. 14, Greenwich CT: JAI Press.
- von Krogh, G., Nonaka, I. & Aben, M. (2001), 'Making the most of your company's knowledge: A strategic framework', *Long Range Planning* **34**(4), 421 – 439.
- Wasserman, S. & Faust, K. (1994), *Social Network Analysis: Methods and Applications*, cambridge University Press.
- Young, L., Simoff, S., Denize, S. & Nankani, E. (2008), Who collaborates with whom and why? exploring the typography and evolution of collaborative networks. IMP Conference 2008, "Studies on business interaction? Consequences for business in theory and business in practice".

Kernel-based Principal Components Analysis on Large Telecommunication Data

Takeshi Sato¹Bingquan Huang¹Guillem Lefait¹M-T. Kechadi¹B. Buckley²¹ School of Computer Science and Informatics

University College Dublin,

Belfield, Dublin 4, Ireland

Email: [bingquan.huang, guillem.lefait, tahar.kechadi]@ucd.ie
takeshi.sato@ucdconnect.ie² Eircom Limited

1 Heuston South Quarter,

Dublin 8, Ireland

Abstract

Linear Principal Components Analysis (LPCA) is known for its simplicity to reduce the features dimensionality. An extension of LPCA, Kernel Principal Components Analysis (KPCA), outperforms LPCA when applied on non-linear data in high dimensional feature space. However, on large datasets with high input space, KPCA deals with a memory issue and imbalance classification problems with difficulty. This paper presents an approach to reduce the complexity of the training process of KPCA by condensing the training set with sampling and clustering techniques as pre-processing step. The experiments were carried out on a large real-world Telecommunication dataset and were assessed on a churn prediction task. The experiments show that the proposed approach, when combined with clustering techniques, can efficiently reduce feature dimension and outperforms standard PCA for customer churn prediction.

Keywords: Kernel PCA, Churn Prediction, Clustering, Imbalanced Classification

1 Introduction

Due to the advances in data collection and storage capability, companies can record considerable amount of information on customers. However, the number of available information is so massive, that both automatic tools and experts face difficulties to analyse these data. Therefore, Dimension Reduction techniques have been developed to select the most adequate information out from high dimensional datasets. In telecommunication service sector, predicting customer churn has become a major focus for companies as churn can result in a huge financial loss. However, analyzing customer data can be troublesome as it contains substantial size of information. One solution is the Feature Extraction (FE) approach.

The main goal of FE approach is to discard redundant attributes and create a new set of attributes that captures the important information more effectively. The FE approach can be subdivided

into three categories: Filter Approach, Heuristic Technique (Genetic Algorithm etc.) and Feature Transformation Approach. The last approach is applied in this paper as it has the advantages of exploring feature combination more effectively.

The Kernel Principal Component Analysis (KPCA) (Schölkopf, Smola & Müller 1998, Schölkopf, Mika, Burges, Knirsch, Müller, Rätsch & Smola 1998) is a renowned feature transformation approach that transforms the original features into new features with orthogonal transformations based on eigenvectors. KPCA is the extension of Linear Principal Component Analysis (LPCA) (I.T.Jolliffe 1986) which extends LPCA by mapping the input data into non-linear feature space and operates PCA with the support of Kernel trick. However, the computational complexity and the memory size required during the training process of KPCA depend on the size of the training dataset. Following issues are expected when applying KPCA on Telecommunication Data:

- Standard KPCA may not be a suitable solution to transform a very large dataset due to its high complexity and memory requirements.
- Similarly to LPCA, KPCA has very low discriminant ability to solve imbalanced classification problems as in LPCA (Chawla et al. 2004). Imbalanced classification is present when churn prediction is applied on Telecommunication data as 95% of the customers are non-churners and 5% are churners.

Several approaches have been proposed to solve these limitations by focusing on the simplification of the KPCA training procedure (Franc 2003, Kim 2005, Marukatat 2006). Kim et al. implement iterative KPCA by applying Generalised Hebbian Algorithm (GHA), which is a linear neural network model for unsupervised learning. The author kernelizes GHA to obtain a memory efficient approximation of KPCA and thus its training process can be achieved without storing a large kernel matrix.

Frank et al. (Franc 2003) introduces a simple greedy algorithm to iteratively add input vectors into a new training dataset until the prescribed limits are reached. These limits for input vector selection are the maximum number of vectors to store and the maximum approximation error rate ϵ . This approach reduces the workload of training process because it reduces the size of the training set, thus decrease the memory requirement.

Copyright ©2009, Australian Computer Society, Inc. This paper appeared at the Eighth Australasian Data Mining Conference (AusDM 2009), Melbourne, Australia. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 101, Paul J. Kennedy, Kok-Leong Ong and Peter Christen, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

In (Marukatat 2006), a clustering algorithm called Kernel K-Means is applied on a training dataset to simplify the training process. It applies K-Means clustering on input data after mapping them onto high dimensional feature space.

However, these approaches have not been tested on a real application with a large dataset such as telecommunication data.

In this paper, we focus on condensation method through clustering and sampling techniques on training data before mapping them onto high dimensional feature space. The proposed approach employs a clustering/sampling algorithm to reduce the size of a training dataset, use the reduced dataset to train KPCA and then use the trained KPCA to reduce the feature dimensions (i.e. feature transformation). This process is validated by using a real-world dataset collected from Telecommunication Company.

The rest of this paper is organised as follows: the next section outlines details of the proposed KPCA approach. Experimental results along with discussion are presented in Section 4 and we conclude and highlight some future directions in Section 5.

2 Kernel PCA

KPCA is a feature extraction algorithm which combines the operations of LPCA approach and the Kernel trick technique (Schölkopf, Smola & Müller 1998) to transform data. LPCA has been known for its simplicity and minimal effort for dimension reduction due to the assumption of linearity. However, it is only limited to re-expressing a data as a linear combination of its basis vectors. KPCA extends LPCA by mapping the features into the high dimensional feature space F , which enables KPCA to separate non-linear data more clearly than LPCA that uses a linear combination. The procedure of KPCA feature extraction is explained in the followings.

Consider a dataset $\{x_{i,i=1,2,\dots,N}\}$ with dimensionality d . In order to find the separability of nonlinear data, the KPCA maps the data into a high (possibly infinite) dimensional feature space by using the Kernel trick, and then the KPCA solves the following eigenvalue decomposition problem:

$$\lambda\alpha = \mathbf{K}\alpha, \text{ subject to } \|\alpha\|_2 = \frac{1}{\lambda} \quad (1)$$

\mathbf{K} is the kernel matrix, defined as:

$$\mathbf{K} = \begin{bmatrix} k_{11} & k_{12} & \cdot & \cdot & \cdot & k_{1N} \\ k_{21} & k_{22} & \cdot & \cdot & \cdot & k_{2N} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ k_{N1} & k_{N2} & \cdot & \cdot & \cdot & k_{NN} \end{bmatrix} \quad (2)$$

The element k_{ij} of the kernel matrix can be computed by

$$k_{ij} = \mathbf{k}(x_i, x_j) \quad (3)$$

where $\mathbf{k}(\cdot)$ is a kernel function. One of the most widely used kernel functions is the Gaussian kernel: $\mathbf{k}(x, y) = \exp(-\frac{\|x-y\|^2}{2\sigma^2})$, $\sigma > 0$. Once the Eigenvalue Decomposition problem (See Eq.1) is solved, the

eigenvalues and eigenvectors can be used to project a test data x by:

$$\pi_k(x) = \sum_{i=1}^N \alpha_i^k \mathbf{k}(x_i, x) \quad (4)$$

However, the described KPCA is considered only when the kernel matrix \mathbf{K} is centred. The kernel matrix \mathbf{K} is not centred in general case. To solve this problem, \mathbf{K} needs to be updated by the following equation:

$$\tilde{\mathbf{k}}(x_i, x_j) = k_{ij} - \frac{1}{N} \sum_{a=1}^N k_{ia} - \frac{1}{N} \sum_{a=1}^N k_{aj} + \frac{1}{N^2} \sum_{a,b=1}^N k_{ab} \quad (5)$$

Similarly, the data projection for this case is performed by

$$\pi_k(x) = \frac{1}{\sqrt{\lambda_k}} \sum_{i=1}^N \alpha_i^k \tilde{\mathbf{k}}(x_i, x) \quad (6)$$

where λ_k is the k^{th} largest non-zero Eigenvalue, α_i^k denotes i^{th} value of λ_k 's corresponding Eigenvector and x_{test} is a test data.

In summary, following steps are required in order to obtain the principal components in KPCA (Schölkopf, Smola & Müller 1998): Firstly, decide which Kernel function to apply and then compute Kernel matrix \mathbf{K} (Eq.2); secondly solve Eigenvalue problem; thirdly project the test data onto the Eigenvectors (Eq.6).

3 Proposed Approach

3.1 KPCA applied to telecommunication data

When KPCA is applied to a dataset of N input data, a kernel matrix of size $N \times N$ is required to be computed. This matrix is required in both the training and the data projection parts. With increasing size of input data to train KPCA, the size of kernel matrix increases. This leads to an issue in the memory requirements and in the computational complexity. In order to solve these problems, sampling and clustering algorithms are applied on a large training dataset to create a dataset of limited size. Since the size of the new dataset is small, the kernel matrix becomes small and thus the computational complexity of solving Eq.1 and the memory to store this kernel matrix are greatly reduced.

The main objective of this paper is to find a solution to avoid KPCA high memory consumption. In order to evaluate our modified KPCA approach in pre-processing step, it is applied on a dataset from Telecommunication services. The accuracy of predicting churn customer is observed and used to compare the solutions. The experiments for this paper are conducted to observe: 1) the comparison of condensation methods, 2) the impact of size of training data and 3) KPCA versus LPCA.

We present an algorithm, a modified KPCA in pre-processing step, described in Algorithm 1 that reduce the size of the training set before applying KPCA. Two approaches are described in Algorithm 1: the class-based, which takes class label into account and the approach that is contrary to class-based. The latter approach selects data based on

Algorithm 1 Adapted KPCA to telecommunication with/without supervision

1. Divide S^{tr} into groups $\{S_i^{tr}, i = 1, \dots, M\}$ according to the class labels, where M is the total number of different labels.
2. For $i=1$ to M ,
 - 2.1 Use either a clustering (e.g. K-Means algorithm) or a sampling algorithm to S_i^{tr} to obtain K number of data, where K is the number of subsets after clustering $\{d_k, k = 1, \dots, K\}$.
 - 2.2 For $k = 1$ to K ,
 - 2.2.1 For sampling select an instance x_{new}^k from S_i^{tr} randomly.
For clustering, calculate the centre of mass of cluster d_k by

$$x_{new}^k = \frac{\sum_{l=1}^L x_l}{L} \quad (7)$$
 where L is the size of d_k and x_l is a instances of d_k .
 - 2.2.2 Add x_{new}^k to S_i^{new} .
 - 2.3 Store instances in S_i^{new} dataset into S^{new}
3. Use S^{new} to train KPCA by the training process described in section 2.
4. Apply trained KPCA to project the datasets S^{new} and S^{te} by Eq. (6).

condensation methods without considering the class label (churn or non-churn). There is a possibility that this approach leads to imbalanced classification problems due to disproportion in the number of class labels. The former approach, on the contrary, solves previous approach limitations by creating a dataset S^{new} that contains classes with same proportion (If $K = 1024$, 512 churn and 512 non-churn are allocated to form S^{new}). For the latter approach of Algorithm 1, initialise the algorithm from Step 2.1 as first step and remove Step 1 and Step 2 iteration. Replace S_i^{tr} and S_i^{new} as S^{tr} and S^{new} . On the other hand, Step 1 and Step 2 iteration are added for the class-based approach so that the size of churn and non-churn in S^{new} are due proportion.

Figure 1 illustrates the workflow of the experiments. The data provided in Step 1 is the original training dataset, S^{tr} , which contains 100,000 customers (non-churners and churners) in the ratio of 19:1. Clustering/Sampling techniques are then applied to this dataset to generate a reduced training dataset, S^{new} , of size K . In Step 3 the reduced dataset is used to train KPCA and transform the test dataset, S^{te} and possibly S^{tr} . The α in Step 3 refers to the number of attributes in transformed dataset, which is dependent on threshold parameter. These two datasets are employed for classification purposes afterwards. In addition, the transformed dataset can be visualized in Step 4.

3.2 DataSet Description

139,000 customers were randomly selected from a real world database provided by Eircom for the experiments. The training set is composed of 6,000 churners and 94,000 non-churners. The test dataset contains 39,000 customers (2,000 churners and 37,000 non-churners). These data contain 122 features which describe individual customer characteristics. Following items describe features that are considered for customer churn prediction problem. See (Huang et al. 2009) for more detailed descriptions:

- Broadband Internet and telephone line information
- Broadband monthly usage information
- Demographic profiles
- Information of grants

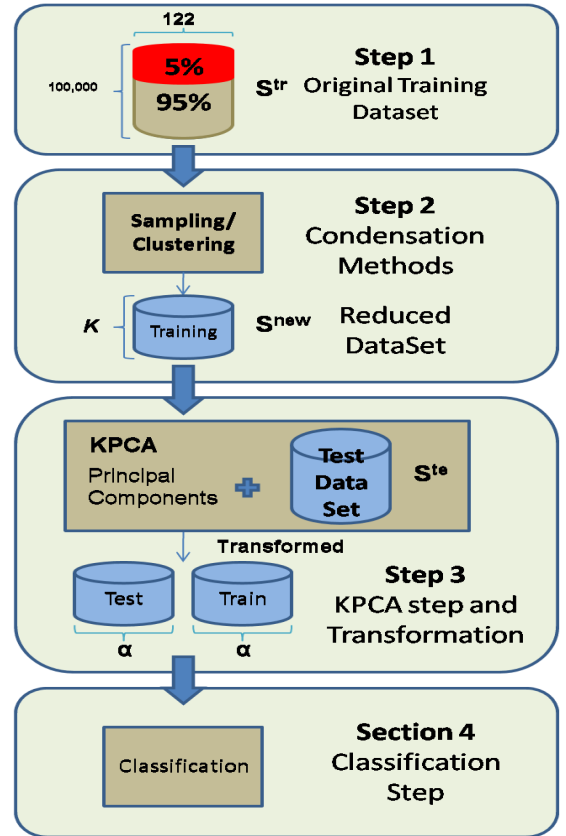


Figure 1: Workflow of the algorithm

- Account Information
- Service orders
- Henley Segments
- The historical information of payments and bills
- Call details

For example, features from *Broadband Internet and telephone line information* could give additional information as to why customer cease contract for the provided services. It includes information about voice

mail service (provided or not), the number of broadband lines, the number of telephone lines, and so on. Due to confidentiality of Customer data, full details of each attribute sets cannot be explained.

3.3 Condensation Methods

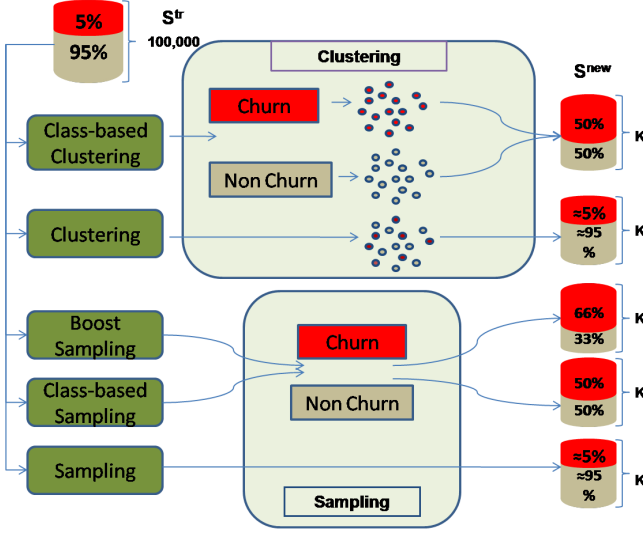


Figure 2: Condensation methods used in the experiments

Standard approach and Class-based approaches are applied on clustering and sampling techniques. Standard approach selects K cluster centre points but neglects the proportion of each class label. On the other hand, class-based approach takes the class label into account and therefore keeps a proportion of 1:1 between churners and non-churners.

There are a total of 5 different condensation methods as shown in Fig.2. First and second sets are Class-based and Standard clustering. In the clustering approach S^{tr} , the original imbalanced training dataset, is given to the K-Means algorithm, and is clustered into K groups. In Class-based Clustering, S^{tr} is separated into number of datasets based on the class label and clustering technique is applied on each subset. Although clustering a dataset may be computational expensive, this can be solved by less complex version of K-Means such as on-line K-Means.

In order to compare the interest of clustering-based condensation method, we compare it to usual sampling methods. The last three condensation methods are random Sampling, Boosted Sampling, Class-based Sampling. The first sampling method is a conventional method, which randomly selects K customers without any condition which leads to unknown size of churners and non-churners. On the contrary, the latter two methods conditionally select data. The class-based Sampling method selects equal number of customer for each class randomly to keep the ratio to 1:1. In Boosted sampling, which is also a class-based sampling, we increase the ratio of churners vs. non-churners to 2:1. Following the pre-processing step, we apply the newly generated dataset S^{new} to KPCA.

For both clustering and sampling approaches K is set to be in $\{2, 4, 8, 16, 32, 64, 128, 256, 512\}$ where K is the number of customers in S^{new} , the dataset that will be given to the KPCA process.

In order to avoid biased classification 10 datasets were generated for each method for each K . These datasets are then given to KPCA to generate transformed datasets and used for churn prediction task. Following the task, the results of each dataset of same method are averaged for comparison purposes.

3.4 KPCA Configuration

Gaussian Radial Basis Function (RBF) is selected as the kernel function in the experiments. This method requires a parameter, the Gaussian width σ . The parameter σ is set to 46 throughout the experiments. The number of PC to keep can be decided either by specifying the exact number of features or setting a threshold. In the experiments, we set 0.9 as the threshold, which means that eigenvectors that account for 90% of total eigenvalues after solving Eq.1 are kept for feature projection.

The two parameters described earlier were employed based on previous experiments with smaller datasets and with different classifiers. However, the setting of these parameters should be assessed directly from the transformed dataset after KPCA.

3.5 Evaluation Criteria

According to the literature and previous experiments in (Huang et al. 2009), the Decision Tree C4.5 (R. 1996, 1993) performed the best results among other classifiers (SVM, Neural Networks) when applied on telecommunication data. Therefore, the Decision Tree C4.5 was employed.

The performance of the predictive churn model has to be evaluated. Table 1 shows a confusion matrix, where a_{11} , resp. a_{22} is the number of the correctly predicted churners, resp. non-churners, and a_{12} , resp. a_{21} is the number of the incorrectly predicted churners, resp. non-churners. Following evaluation criteria

		Predicted	
		CHU	NONCHU
Actual	CHU	a_{11}	a_{12}
	NONCHU	a_{21}	a_{22}

Table 1: Confusion Matrix

are used in the experiments (Hamilton et al. n.d.);

- The accuracy of true churn (TP) is defined as the proportion of churn cases that were classified correctly: $TP = \frac{a_{11}}{a_{11} + a_{12}}$.
- The false churn rate (FP) is the proportion of non churn cases that were incorrectly classified as churn: $FP = \frac{a_{21}}{a_{21} + a_{22}}$.

A good solution should have both a high TP and a low FP. When comparing two solutions, S_1 and S_2 , if S_1 'TP is above S_2 'TP and S_1 'FP is below S_2 'FP, S_1 is considered to dominate S_2 and is considered as the best solution. When no solution is dominant, the evaluation depends on the expert strategy, i.e. to favour TP or FP.

4 Results and Discussion

Figure.3 presents the averaged results and the deviation of each condensation methods with a S^{new} of different size. Among the condensation methods, simple sampling produced the lowest churn prediction results with a maximum of 63% at $K=256$, resp. 62% at $K=512$, as shown in Fig.3(a). In Fig.3(b), the class-based sampling methods performed better

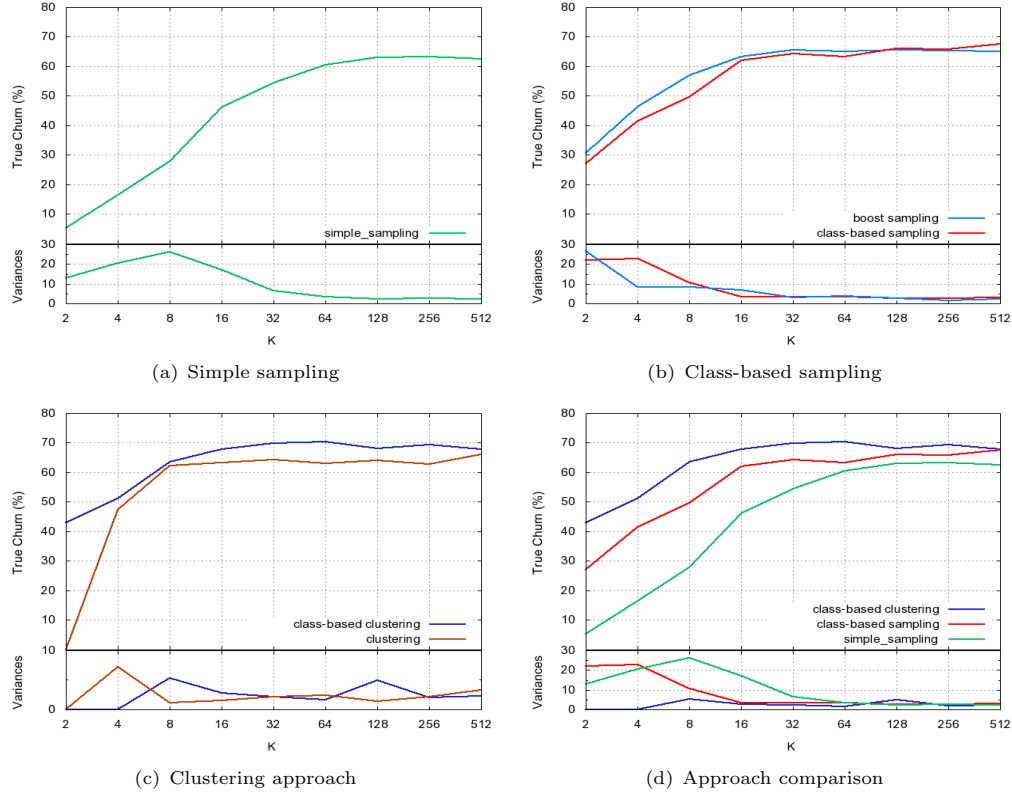


Figure 3: TP and variance for diverse condensation methods with varying size

with 68% and 66% as the highest TP for class-based (1:1) and boost (2:1) sampling respectively. Although the boost sampling produced similar results to class-based sampling, increasing the number of churn in S^{new} do not increase the TP accuracy. The class-based clustering approach recorded the highest TP rate among other condensation methods with 70% at $K=64$ shown in Fig.3(c). On the other hand, standard clustering produced similar results to class-based sampling methods. Fig.3(d) compares the best approach from each previous figure to highlight the difference between sampling and clustering approaches. It includes simple and class-based sampling and class-based clustering methods.

The curve line in each graph in Fig.3 indicate that it is not necessary to consider all the data to obtain maximum accuracy. On all approaches after a given K , results tend to be of same order.

	LPCA	Class.Clus	Simple.Samp	Class.Samp
TP (%)	69.35	73.45	67	71.65
FP (%)	4.05	1.703	2.56	2.83

Table 2: The best Prediction rates vs. LPCA and the modified KPCA approach

Table 2 compares the prediction rates between the adapted KPCA and LPCA and Figure 4 plots the first PC against the second PC following the KPCA projections for 2D visualization. The dataset that produced the best results out of 10 datasets from each condensation methods in Fig.3 were used as training set and compared. The test set used for Figure 5 is described in Section 3.2. The red and green dots refer to churners and non-churners respectively. We can see that class-based clustering approach produced the best results with best separability of classes among the 4 approaches. Fig.5 plots the receiver operating characteristics (ROC) of TP against FP. It shows

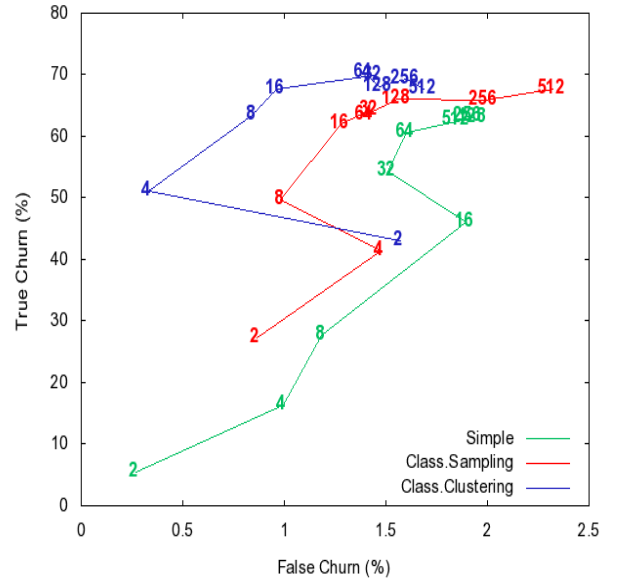


Figure 5: ROC graph: True Churn vs. False Churn for different approaches

that the values of class-based clustering is above of both simple and class-based sampling (it dominates them except for $K=2$). Class-based Clustering with $K=16$ has TP of around 70% and FP of below 1%. The remaining 30% of TP is classified as non churner while being churner. Similarly, the remaining 99% of FP is classified correctly as non churner. Therefore class-based clustering approach is the best approach among the condensation methods: *class-based clustering* dominates *clustering* and *clustering* dominates *sampling*.

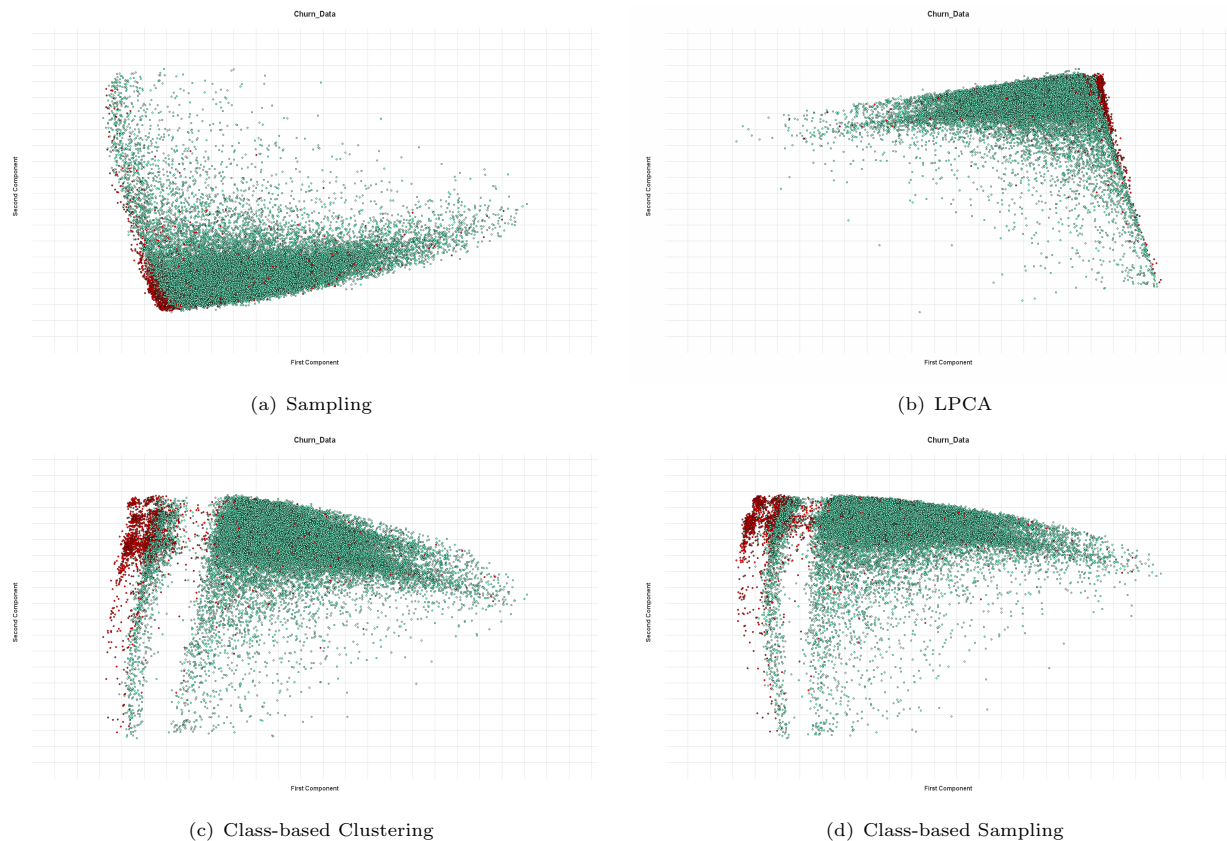


Figure 4: 2D visualization for different approaches after KPCA/LPCA

In summary, the experiments showed that clustering approach produced a more adequate summary of the dataset than sampling approach and leads to a better prediction rate both in TP and FP. To conclude, KPCA with clustering approach in pre-processing step is more efficient than LPCA and sampling approach.

5 Conclusions and Future Perspective

In this paper, we have proposed a modified KPCA approach to be able to process large datasets and to improve performances on imbalanced classification problem. The proposed KPCA approach focuses on reducing the memory requirements of the training process in the KPCA by reducing the size of the training dataset through sampling and clustering techniques. To solve KPCA limitations, Kernel matrix size is reduced as it accounts for most memory space in KPCA and the number of each class in a training dataset is adjusted for balanced classification purpose. Condensing techniques used were simple and class-based sampling and K-Means clustering.

The modified approach was tested on a telecommunication dataset on a churn prediction task. The results show that KPCA with clustering approach outperformed LPCA and sampling approach. In addition to this, the approach that takes class label into account performed better in both clustering and sampling.

However, there are some problems in using the proposed approach. Class-based clustering algorithm produced the best results but we need to investigate

if a different clustering technique will produce better results than K-Means. We plan to use other clustering algorithms such as methods based on density, like DBSCAN, or based on subspaces, like CLIQUE.

Another limitation of using KPCA is the evaluation of the transformed features. It should be assessed directly without relying on the classifier results, because this evaluation is very costly. Therefore, we will measure and assess the separability of classes to be able to select internally the most suitable feature transformation.

Finally, the right parameters should be discovered automatically. In this paper, we set the Gaussian kernel width as 46 and the number of PC with a threshold of 0.9 based on previous experiments. In future works, we plan to develop a method to estimate automatically these parameters.

References

- Chawla, N. V., Japkowicz, N. & Kotcz, A. (2004), 'Editorial: special issue on learning from imbalanced data sets', *SIGKDD Explor. Newsl.* **6**(1), 1–6.
- Franc, V., V. (2003), Greedy algorithm for a training set reduction in the kernel methods, in '10th International Conference on Computer Analysis of Images and Patterns', Groningen, The Netherlands, pp. 426–433.

This research was partly supported by Eircom of Ireland.

- Hamilton, H., Gurak, E., Findlater, L., Olive, W. & Ranson, J. (n.d.). http://www2.cs.uregina.ca/~dbd/cs831/notes/confusion_matrix/confusion_matrix.html.
- Huang, B., Kechadi, M.-T. & Buckley, B. (2009), Customer churn prediction for broadband internet services, in '11th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2009)', LNCS 5691, Linz, Austria, pp. 229–243.
- I.T.Jolliffe (1986), *Principal Components Analysis*, Springer-Verlag.
- Kim, K.I., F. M. S. B. (2005), 'Iterative kernel principal component analysis for image modelling', *IEEE Transactions on Pattern Analysis and Machine Intelligence* pp. 1351–1366.
- Marukatat, S. (2006), 'Sparse kernel pca by kernel k-means and preimage reconstruction algorithms', *Trends in Artificial Intelligence* **Volume 4099/2006**, 454–463.
- R., Q. J. (1993), 'C4.5: Programs for machine learning'.
- R., Q. J. (1996), 'Improved use of continuous attributes in c4.5', *Journal of Artificial Intelligence Research* **4**, 77–90.
- Schölkopf, B., Mika, S., Burges, C., Knirsch, P., Müller, K.-R., Rätsch, G. & Smola, A. (1998), 'Input space versus feature space in kernel-based methods', *IEEE Transactions on Neural Networks* **5**(10), 10001016.
- Schölkopf, B., Smola, A. & Müller, K. (1998), 'Non-linear component analysis as a kernel eigenvalue problem', *Neural Computation* **5**(10), 1299–13995.

SparseDTW: A Novel Approach to Speed up Dynamic Time Warping

Ghazi Al-Naymat^{1,*}

Sanjay Chawla²

Javid Taheri²

¹ School of Computer Science and Engineering
The University of New South Wales
Sydney, NSW 2052, Australia
Email: ghazi@cse.unsw.edu.au

² School of Information Technologies
The University of Sydney, Australia
Email: {chawla,javid}@it.usyd.edu.au

Abstract

We present a new space-efficient approach, (*SparseDTW*), to compute the Dynamic Time Warping (*DTW*) distance between two time series that always yields the optimal result. This is in contrast to other known approaches which typically sacrifice optimality to attain space efficiency. The main idea behind our approach is to dynamically exploit the existence of similarity and/or correlation between the time series. The more the similarity between the time series the less space required to compute the *DTW* between them. To the best of our knowledge, all other techniques to speedup *DTW*, impose apriori constraints and do not exploit similarity characteristics that may be present in the data. We conduct experiments and demonstrate that *SparseDTW* outperforms previous approaches.

Keywords: Time series, Similarity measures, Dynamic time warping, Data mining

1 Introduction

Dynamic time warping (*DTW*) uses the dynamic programming paradigm to compute the alignment between two time series. An *alignment* “warps” one time series onto another and can be used as a basis to determine the similarity between the time series. *DTW* has similarities to sequence alignment in bioinformatics and computational linguistics except that the *matching* process in sequence alignment and *warping* have to satisfy a different set of constraints and there is no gap condition in warping. *DTW* first became popular in the speech recognition community (Sakoe & Chiba 1978) where it has been used to determine if the two speech wave-forms represent the same underlying spoken phrase. Since then it has been adopted in many other diverse areas and has become the similarity metric of choice in time series analysis (Keogh & Pazzani 2000).

Like in sequence alignment, the standard *DTW* algorithm has $O(mn)$ space complexity where m and n are the lengths of the two sequences being aligned. This limits the practicality of the algorithm in today's “data rich environment” where long sequences are of-

ten the norm rather than the exception. For example, consider two time series which represent stock prices at one second granularity. A typical stock is traded for at least eight hours on the stock exchange and that corresponds to a length of $8 \times 60 \times 60 = 28800$. To compute the similarity, *DTW* would have to store a matrix with at least 800 million entries!

Figure 1(a) shows an example of an alignment (warping) between two sequences S and Q . It is clear that there are several possible alignments but the challenge is to select the one which has the minimal overall distance. The alignment has to satisfy several constraints which we will elaborate on in Section 3.

Salvador & Chan (2007) have provided a succinct categorization of different techniques that have been used to speed up *DTW*:

- **Constraints:** By adding additional constraints the search space of possible alignments can be reduced. Two well known exemplars of this approach are the Sakoe & Chiba (1978) and the Itakura (1975) constraints which limit how far the alignment can deviate from the diagonal. While these approaches provide a relief in the space complexity, they do not guarantee the optimality of the alignment.
- **Data Abstraction:** In this approach, the warping path is computed at a lower resolution of the data and then mapped back to the original resolution (Salvador & Chan 2007). Again, optimality of the alignment is not guaranteed.
- **Indexing:** Keogh & Ratanamahatana (2004), Sakurai et al. (2005), and Lemire (2009) proposed an indexing approach, which does not directly speed up *DTW* but limits the number of *DTW* computations. For example, suppose there exists a database D of time series sequences and a query sequence q . We want to retrieve all sequences $d \in D$ such that $DTW(q, d) < \epsilon$. Then instead of checking q against each and every sequence in D , an easy to calculate lower bound function LBF is first applied between q and D . The argument works as follows:

1. By construction, $LBF(q, d) < DTW(q, d)$.
2. Therefore, if $LBF(q, d) > \epsilon$ then $DTW(q, d) > \epsilon$ and $DTW(q, d)$ does not have to be computed.

1.1 Main Contribution

The main insight behind our proposed approach, *SparseDTW*, is to dynamically exploit the possible

*The work was done while the author at School of Information Technologies, The University of Sydney.
Copyright ©2009, Australian Computer Society, Inc. This paper appeared at the Eighth Australasian Data Mining Conference (AusDM 2009), Melbourne, Australia. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 101. Paul J. Kennedy, Kok-Leong Ong, and Peter Christen, Eds. Reproduction for academic, not-for profit purposes permitted provided this text is included.

existence of inherent similarity and correlation between the two time series whose *DTW* is being computed. This is the motivation behind the Sakoe-Chiba band and the Itakura Parellelogram but our approach has three distinct advantages:

1. Bands in *SparseDTW* evolve dynamically and are, on average, much smaller than the traditional approaches. We always represent the warping matrix using sparse matrices, which leads to better average space complexity compared to other approaches (Figure 9).
2. *SparseDTW* always yields the optimal warping path since we never have to set apriori constraints independently of the data. For example, in the traditional banded approaches, a sub-optimal path will result if all the possible optimal warping paths have to cross the bands.
3. Since *SparseDTW* yields an optimal alignment, it can easily be used in conjunction with lower bound approaches.

1.2 Paper Outline

The rest of the paper is organized as follows: Section 2 describes related work on *DTW*. The *DTW* algorithm is described in Section 3. In Section 4, we give an overview of the techniques used to speed up *DTW* by adding constraints. Section 5 reviews the Divide and Conquer approach for *DTW* which is guaranteed to take up $O(m+n)$ space and $O(mn)$ time. Furthermore, we provide an example which clearly shows that the divide and conquer approach fails to arrive at the optimal *DTW* result. The *SparseDTW* algorithm is introduced with a detailed example in Section 6. We analyze and discuss our results in Section 7, followed by our conclusions in Section 8.

2 Related Work

DTW was first introduced in the data mining community in the context of mining time series (Berndt & Clifford 1994). Since it is a flexible measure for time series similarity it is used extensively for ECGs (Electrocardiograms) (Caiani et al. 1998), speech processing (Rabiner & Juang 1993), and robotics (Schmill et al. 1999). It is important to know that *DTW* is a measure not a metric, because *DTW* does not satisfy the triangular inequality.

Several techniques have been introduced to speed up *DTW* and/or reduce the space overhead (Hirschberg 1975, Yi et al. 1998, Kim et al. 2001, Keogh & Ratanamahatana 2004, Lemire 2009).

Divide and conquer (DC) heuristic proposed by Hirschberg (1975); that is a dynamic programming algorithm that finds the least cost sequence alignment between two strings in linear space and quadratic time. The algorithm was first used in speech recognition area to solve the Longest Common Subsequence (LCSS). However as we will show with the help of an example, *DC* does not guarantee the optimality of the *DTW* distance.

Sakoe & Chiba (1978) speed up the *DTW* by constraining the warping path to lie within a band around the diagonal. However, if the optimal path crosses the band, the result will not be optimal.

Keogh & Ratanamahatana (2004) and Lemire (2009) introduced efficient lower bounds that reduce the number of *DTW* computations in a time series database context. However, these lower bounds do not reduce the space complexity of the *DTW* computation, which is the objective of our work.

Sakurai et al. (2005) presented FTW, a search method for *DTW*; it adds no global constraints on *DTW*. Their method designed based on a lower bounding distance measure that approximates the *DTW* distance. Therefore, it minimizes the number of *DTW* computations but does not increase the speed the *DTW* itself.

Salvador & Chan (2007) introduced an approximation algorithm for *DTW* called *FastDTW*. Their algorithm begins by using *DTW* in very low resolution, and progresses to a higher resolution linearly in space and time. *FastDTW* is performed in three steps: coarsening shrinks the time series into a smaller time series; the time series is projected by finding the minimum distance (warping path) in the lower resolution; and the warping path is an initial step for higher resolutions. The authors refined the warping path using local adjustment. *FastDTW* is an approximation algorithm, and thus there is no guarantee it will always find the optimal path. It requires the coarsening step to be run several times to produce many different resolutions of the time series. The *FastDTW* approach depends on a radius parameter as a constraint on the optimal path; however, our technique does not place any constrain while calculating the *DTW* distance.

DTW has been used in data streaming problems. Capitani & Ciaccia (2007) proposed a new technique, Stream-*DTW* (*STDW*). This measure is a lower bound of the *DTW*. Their method uses a sliding window of size 512. They incorporated a band constraint, forcing the path to stay within the band frontiers, as in (Sakoe & Chiba 1978).

All the above algorithms were proposed either to speed up *DTW*, by reducing its space and time complexity, or reducing the number of *DTW* computations. Interestingly, the approach of exploiting the similarity between points (correlation) has never, to the best of our knowledge, been used in finding the optimality between two time series. *SparseDTW* considers the correlation between data points, that allows us to use a sparse matrix to store the warping matrix instead of a full matrix. We do not believe that the idea of sparse matrix has been considered previously to reduce the required space.

Algorithm 1 *DTW*: The standard *DTW* algorithm.

Input: S : Sequence of length n , Q : Sequence of length m .

Output: *DTW* distance.

- 1: Initialize $D(i, 1) \leftarrow i\delta$ for each i
 - 2: Initialize $D(1, j) \leftarrow j\delta$ for each j
 - 3: **for all** i such that $2 \leq i \leq n$ **do**
 - 4: **for all** j such that $2 \leq j \leq m$ **do**
 - 5: Use Equation 3 to compute $D(i, j)$
 - 6: **end for**
 - 7: **end for**
 - 8: **return** $D(n, m)$
-

3 Dynamic Time Warping (DTW)

DTW is a dynamic programming technique used for measuring the similarity between any two time series with arbitrary lengths. This section gives an overview of *DTW* and how it is calculated. The following two time series (Equations 1 and 2) will be used in our explanations.

$$S = s_1, s_2, s_3, \dots, s_i, \dots, s_n \quad (1)$$

$$Q = q_1, q_2, q_3, \dots, q_j, \dots, q_m \quad (2)$$

Where n and m represent the length of time series

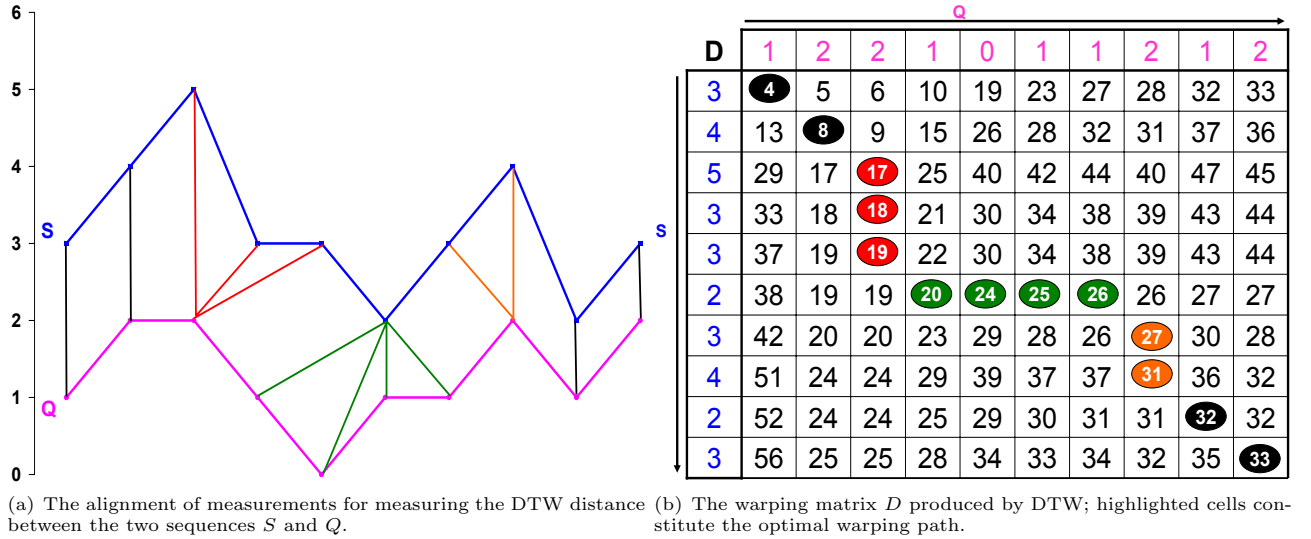


Figure 1: Illustration of DTW.

S and Q , respectively. i and j are the point indices in the time series.

DTW is a time series association algorithm that was originally used in speech recognition (Sakoe & Chiba 1978). It relates two time series of feature vectors by warping the time axis of one series onto another.

As a dynamic programming technique, it divides the problem into several sub-problems, each of which contribute in calculating the distance cumulatively. Equation 3 shows the recursion that governs the computations is:

$$D(i, j) = d(i, j) + \min \begin{cases} D(i-1, j) \\ D(i-1, j-1) \\ D(i, j-1) \end{cases} \quad (3)$$

The first stage in the DTW algorithm is to fill a local distance matrix d . That matrix has $n \times m$ elements which represent the Euclidean distance between every two points in the time series (i.e., distance matrix). In the second stage, it fills the warping matrix D (Figure 1(b)) on the basis of Equation 3. Lines 1 to 7 in Algorithm 1 illustrate the process of filling the warping matrix. We refer to the cost between the i^{th} and the j^{th} elements as δ as mentioned in line 1 and 2.

After filling the warping matrix, the final stage for the DTW is to report the optimal warping path and the DTW distance. Warping path is a set of adjacent matrix elements that identify the mapping between S and Q . It represents the path that minimizes the overall distance between S and Q . The total number of elements in the warping path is K , where K denotes the normalizing factor and it has the following attributes:

$$W = w_1, w_2, \dots, w_K$$

$$\max(|S|, |Q|) \leq K < (|S| + |Q|)$$

Every warping path must satisfy the following constraints (Keogh & Ratanamahatana 2004, Salvador & Chan 2007, Sakoe & Chiba 1978):

1. **Monotonicity:** Any two adjacent elements of the warping path W , $w_k = (w_i, w_j)$ and $w_{k-1} = (w'_i, w'_j)$, follow the inequalities, $w_i - w'_i \geq 0$ and

$w_j - w'_j \geq 0$. This constraint guarantees that the warping path will not roll back on itself. That is, both indexes i and j either stay the same or increase (they never decrease).

2. **Continuity:** Any two adjacent elements of the warping path W , $w_k = (w_i, w_j)$ and $w_{k+1} = (w'_i, w'_j)$, follow the inequalities, $w_i - w'_i \leq 1$ and $w_j - w'_j \leq 1$. This constraint guarantees that the warping path advances one step at a time. That is, both indexes i and j can only increase by at most 1 on each step along the path.
3. **Boundary:** The warping path starts from the top left corner $w_1 = (1, 1)$ and ends at the bottom right corner $w_K = (n, m)$. This constraint guarantees that the warping path contains all points of both time series.

Although there are a large number of warping paths that satisfy all of the above constraints, DTW is designed to find the one that minimizes the warping cost (distance). Figures 1(a) and 1(b) demonstrate an example of how two time series (S and Q) are warped and the way their distance is calculated. The circled cells show the optimal warping path, which crosses the grid from the top left corner to the bottom right corner. The DTW distance between the two time series is calculated based on this optimal warping path using the following equation:

$$DTW(S, Q) = \min \left\{ \frac{\sqrt{\sum_{k=1}^K W_k}}{K} \right\} \quad (4)$$

The K in the denominator is used to normalize different warping paths with different lengths.

Since the DTW has to potentially examine every cell in the warping matrix, its space and time complexity is $O(nm)$.

4 Global Constraint (BandDTW)

There are several methods that add global constraints on DTW to increase its speed by limiting how far the warping path may stray from the diagonal of the warping matrix (Tappert & Das 1978, Berndt & Clifford 1994, Myers et al. 1980). In this paper we use

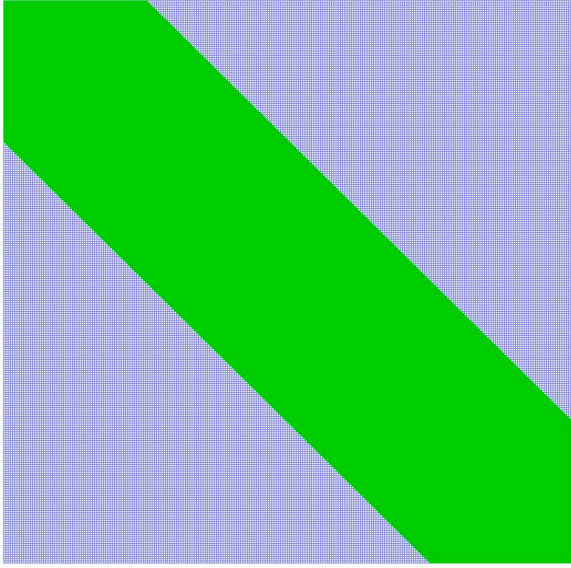


Figure 2: Global constraint (Sakoe Chiba Band), which limits the warping scope. The diagonal green areas correspond to the warping scopes.

Sakoe-Chiba Band (henceforth, we refer to it as Band-DTW) Sakoe & Chiba (1978) when comparing with our proposed algorithm (Figure 2). BandDTW used to speed up the *DTW* by adding constraints which force the warping path to lie within a band around the diagonal; if the optimal path crosses the band, the *DTW* distance will not be optimal.

Algorithm 2 *DC*: Divide and Conquer technique.

Input: S : Sequence of length n , Q : Sequence of length m .

Output: *DTW* distance.

```

1: Divide-Conquer-Alignment( $S, Q$ )
2:  $n \leftarrow |S|$ 
3:  $m \leftarrow |Q|$ 
4:  $Mid \leftarrow \lceil m/2 \rceil$ 
5: if  $n \leq 2$  or  $m \leq 2$  then
6:   Compute optimal alignment using standard DTW
7: else
8:    $f \leftarrow \text{ForwardsSpaceEfficientAlign}(S, Q[1:Mid])$ 
9:    $g \leftarrow \text{BackwardsSpaceEfficientAlign}(S, Q[Mid:m])$ 
10:   $q \leftarrow \text{index that minimizing } f(q, Mid) + g(q, Mid)$ 
11:  Add  $(q, Mid)$  to global array  $P$ 
12:  Divide-Conquer-Alignment( $S[1:q], Q[1:Mid]$ )
13:  Divide-Conquer-Alignment( $S[q:n], Q[Mid:m]$ )
14: end if
15: return  $P$ 

```

5 Divide and Conquer Technique (DC)

In the previous section, we have shown how to compute the optimal alignment using the standard *DTW* technique between two time series. In this section we will show another technique that uses a Divide and Conquer heuristic, henceforth we refer to it as (*DC*), proposed by Hirschberg (1975). *DC* is a dynamic programming algorithm used to find the least cost sequence alignment between two strings. The algorithm was first introduced to solve the Longest Common Subsequence (LCSS) (Hirschberg 1975). Algorithm 2 gives a high level description of the *DC* algorithm. Like in the standard sequence alignment,

the *DC* algorithm has $O(mn)$ time complexity but $O(m + n)$ space complexity, where m and n are the lengths of the two sequences being aligned. We will be using Algorithm 2 along with Figure 3 to explain how *DC* works. In the example we use two sequences $S = [3, 4, 5, 3, 3]$ and $Q = [1, 2, 2, 1, 0]$ to determine the optimal alignment between them. There is only one optimal alignment for this example (Figure 3(e)), where shaded cells are the optimal warping path. The *DC* algorithm works as follows:

1. It finds the middle point in Q which is $Mid = \lfloor Q \rfloor / 2$, (Figure 3(a)). This helps to find the split point which divides the warping matrix into two parts (sub-problems). A forward space efficiency function (Line 8) uses S and the first cut of $Q = [1, 2, 2]$, then a backward step (Line 9) uses S and $Q = [2, 1, 0]$ (Figure 3(a)). Then by adding the last column from the forward and backward steps together and finding the index of the minimum value, the resultant column indicates the row index that will be used along with the middle point to locate the split point (shaded cell in Figure 3(a)). Thus, the first split point is $D(4, 3)$. At this stage of the algorithm, there are two sub-problems; the alignment of $S = [3, 4, 5, 3]$ with $Q = [1, 2, 2]$ and of $S = [3, 3]$ with $Q = [2, 1, 0]$.
2. *DC* is recursive algorithm, each call splits the problem into two other sub-problems if both sequences are of *length* > 2 , otherwise it calls the standard *DTW* to find the optimal path for that particular sub-problem. In the example, the first sub-problem will be fed to Line 12 which will find another split point, because both input sequences are of *length* > 2 . Figure 3(b) shows how the new split point is found. Figure 3(c) shows the two split points (shaded cells) which yield to have sub-problems of sequences of *length* ≤ 2 . In this case *DTW* will be used to find the optimal alignment for each sub-problem.
3. The *DC* algorithm finds the final alignment by concatenating the results from each call of the standard *DTW*.

The example in Figure 3 clarifies that the *DC* algorithm does not give the optimal warping path. Figures 3(d) and (e) show the paths obtained by the *DC* and *DTW* algorithms, respectively.

DC does not yield the optimal path as it goes into infinite recursion because of how it calculates the middle point. *DC* calculates the middle point as follows:

There are two scenarios: first, when the middle point (Algorithm 2 Line 4) is floored ($Mid = \lfloor m/2 \rfloor$) and second when it is rounded up ($Mid = \lceil m/2 \rceil$). The first scenario causes infinite recursion, since the split from the previous step gives the same sub-sequences (i.e., the algorithm keeps finding the same split point). The second scenario is shown in Figures 3(a-d), which clearly confirms that the final optimal path is not the same as the one retrieved by the standard *DTW*¹. The final *DTW* distance is different as well. The shaded cells in Figures 3(d) and (e) show that both warping paths are different.

6 Sparse Dynamic Programming Approach

In this section, we outline the main principles we use in *SparseDTW* and follow up with an illustrated example along with the *SparseDTW* pseudo-code. We exploit the following facts in order to reduce space usage while avoiding any re-computations:

¹It should be noted that our example has only one optimal path that gives the optimal distance.

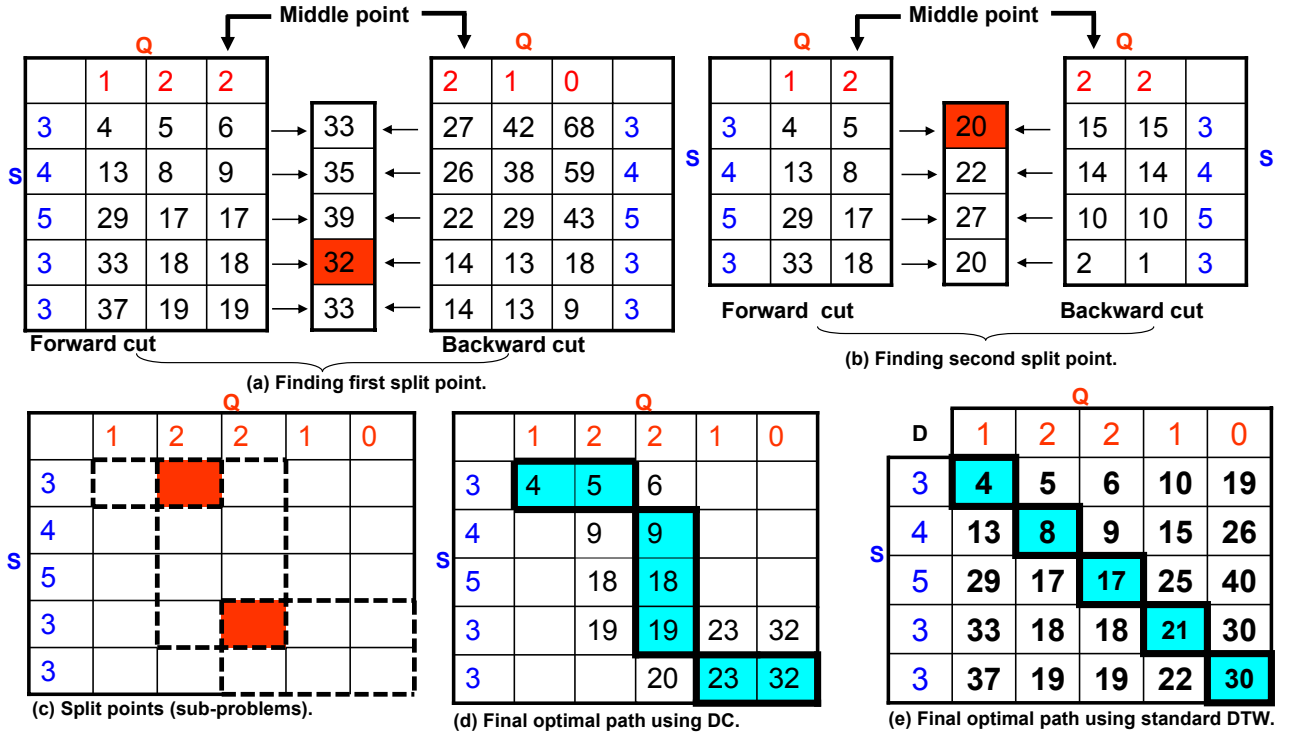


Figure 3: An example to show the difference between the standard DTW and the DC algorithm.

1. Quantizing the input time series to exploit the similarity between the points in the two time series.
2. Using a sparse matrix of size k , where $k = n \times m$ in the worst case. However, if the two sequences are similar, $k \ll n \times m$.
3. The warping matrix is calculated using dynamic programming and sparse matrix indexing.

6.1 Key Concepts

In this section we introduce the key concepts used in our algorithm.

Definition 1 (Sparse Matrix SM) is a matrix that is populated largely with zeros. It allows the techniques to take advantage of the large number of zero elements. Figure 4(a) shows the SM initial state. SM is linearly indexed. The little numbers, in the top left corner of SM 's cells, represent the cell index. For example, the indices of the cells $SM(1,1)$ and $SM(5,5)$ are 1 and 25, respectively.

Definition 2 (Lower Neighbors ($LowerNeighbors$)) a cell $c \in SM$ has three lower neighbors which are the cells of the indices $(c-1)$, $(c-n)$, and $(c-(n+1))$ (where n is the number of rows in SM). For example, the lower neighbors of cell $SM(12)$ are $SM(6)$, $SM(7)$ and $SM(11)$ (Figure 4(a)).

Definition 3 (Upper Neighbors ($UpperNeighbors$)) a cell $c \in SM$ has three upper neighbors which are the cells of the indices $(c+1)$, $(c+n)$, and $(c+n+1)$ (where n is the number of rows in SM). For example, the upper neighbors of cell $SM(12)$ are $SM(13)$, $SM(17)$ and $SM(18)$ (Figure 4(a)).

Definition 4 (Blocked Cell (B)) a cell $c \in SM$ is blocked if its value is zero. The letter (B) refers to the blocked cells (Figure 4(a)).

Definition 5 (Unblocking) Given a cell $c \in SM$, if $SM(c)$'s upper neighbors ($SM(c+1)$, $SM(c+n)$, and $SM(c+n+1)$) are blocked, they will be unblocked. Unblocking is performed by calculating the $EucDist$ for these cells and adding them to SM . In other words, adding the distances to these cells means changing their state from blocked (B) into unblocked. For example, $SM(10)$ is a blocked upper neighbor of $SM(5)$, in this case $SM(10)$ needs to be unblocked (Figure 4(c)).

6.2 SparseDTW Algorithm

The algorithm takes Res , the resolution parameter as an input that determines the number of bins as $\frac{2}{Res}$. Res will have no impact on the optimality. We now present an example of our algorithm to illustrate some of the highlights of our approach: We start with two sequences:

$$S = [3, 4, 5, 3, 3] \text{ and } Q = [1, 2, 2, 1, 0].$$

In Line 1, we first quantize the sequences into the range $[0, 1]$ using Equation 5:

$$QuantizedSeq_i^k = \frac{S_i^k - \min(S^k)}{\max(S^k) - \min(S^k)}. \quad (5)$$

Where S_i^k denotes the i^{th} element of the k^{th} time series. This yields the following sequences:

$$S' = [0, 0.5, 1.0, 0.0, 0.0] \text{ and } Q' = [0.5, 1.0, 1.0, 0.5, 0]$$

In Lines 4 to 7 we create *overlapping* bins, governed by two parameters: bin-width and the overlapping width (which we refer to as the resolution). It is important to note that these two parameters do not affect the optimality of the alignment but do have an affect on the amount of space utilized. For this particular example, the bin-width is 0.5. We thus have 4 bins which are shown in Table 1.

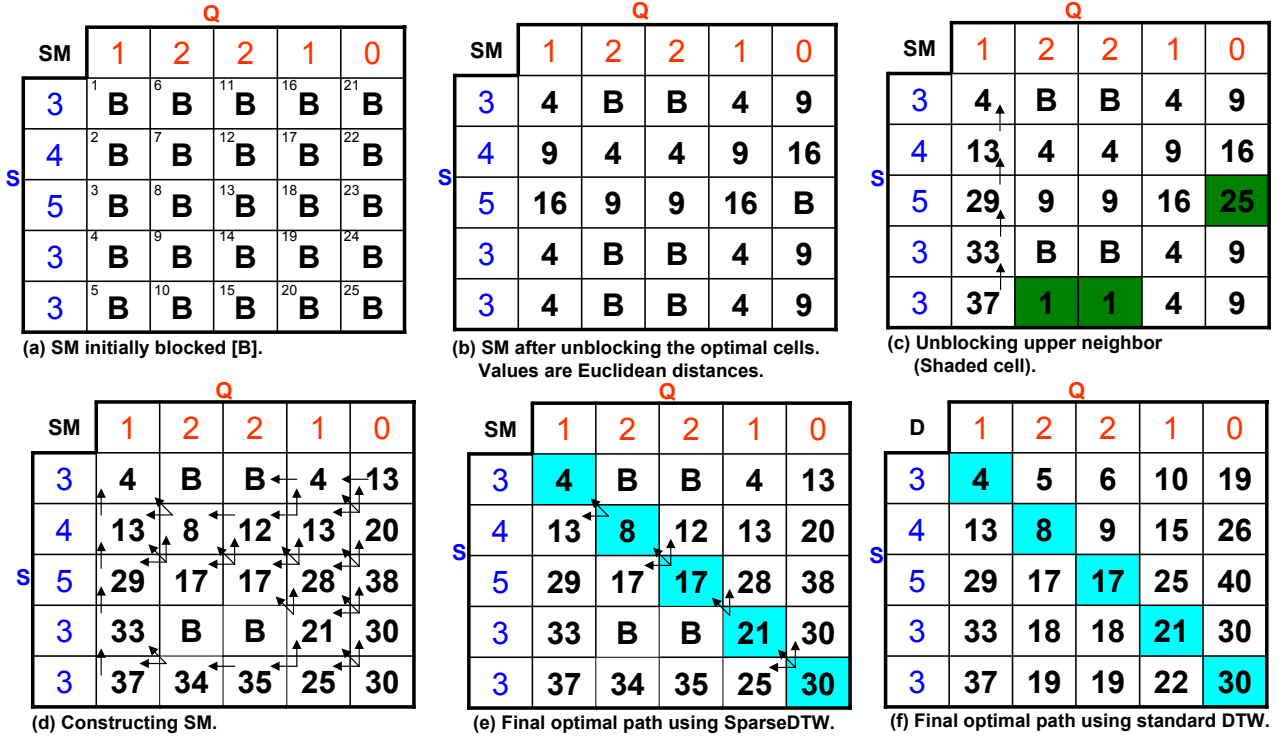


Figure 4: An example of the SparseDTW algorithm and the method of finding the optimal path.

Bin Number (B_k)	Bin Bounds	Indices of S'	Indices of Q'
1	0.0-0.5	1,2,4,5	1,4,5
2	0.25-0.75	2	1,4
3	0.5-1.0	2,3	1,2,3,4
4	0.75-1.25	3	2,3

Table 1: Bins bounds, where B_k is the k^{th} bin.

Our intuition is that points in sequences with similar profiles will be mapped to other points in the same bin or neighboring bins. In which case the non-default entries of the sparse matrix can be used to compute the warping path. Otherwise, default entries of the matrix will have to be “opened”, reducing the sparsity of the matrix but never sacrificing the optimal alignment.

In Lines 3 to 13, the sparse warping matrix SM is constructed using the equation below. SM^2 is a matrix that has generally few non-zero (or “interesting”) entries. It can be represented in much less than $n \times m$ space, where n and m are the lengths of the time series S and Q , respectively.

$$SM(i, j) = \begin{cases} EucDist(S(i), Q(j)) & \text{if } S(i) \text{ and } Q(j) \in B_k \\ B & \text{otherwise} \end{cases} \quad (6)$$

We assume that SM is linearly ordered and the default value of SM cells are zeros. That means the cells initially are *Blocked* (B) (Figure 4(a)). Figure 4(a) shows the linear order of the SM matrix, where the little numbers on the top left corner of each cell represent the index of the cells. In Line 6 and 7, we find the index of each quantized value that falls in the bin bounds (Table 1 column 2, 3 and 4). The Inequal-

²If the Euclidean distance (EucDist) between $S(i)$ and $Q(j)$ is zero, then $SM(i, j) = -1$, to distinguish between a blocked cell and any cell that represents zero distance.

ity 7 is used in Line 6 and 7 to find the indices of the default entries of the SM .

$$LowerBound \leq QuantizedSeq_i^k \leq UpperBound. \quad (7)$$

Where $LowerBound$ and $UpperBound$ are the bin bounds and $QuantizedSeq_i^k$ represents the quantized time series which can be calculated using Equation 5.

Lines 8 to 12 are used to initialize the SM . That is by joining all indices in $idxS$ and $idxQ$ to open corresponding cells in SM . After unblocking (opening) the cells that reflect the similarity between points in both sequences, the SM entries are shown in Figure 4(b).

Lines 14 to 22 are used to calculate the warping cost. In Line 15, we find the warping cost for each open cell $c \in SM$ (cell c is the number from the linear order of SM 's cells) by finding the minimum of the costs of its lower neighbors, which are $[c-1, c-n, c-(n+1)]$ (black arrows in Figure 4(d) show the lower neighbors of every open cell). This cost is then added to the local distance of cell c (Line 17). The above step is similar to DTW , however, we may have to open new cells if the upper neighbors at a given local cell $c \in SM$ are blocked. The indices of the upper neighbors are $[c+1, c+n, c+n+1]$, where n is the length of sequence S (i.e., number of rows in SM). Lines 18 to 21 are used to check always the upper neighbors of $c \in SM$. This is performed as follows: if the $|UpperNeighbors| = 0$ for a particular cell, its upper neighbors will be unblocked. This is very useful when the algorithm traverses SM in reverse to find the final optimal path. In other words, unblocking allows the path to be connected. For example, the cell $SM(5)$ has one upper neighbor that is cell $SM(10)$ which is blocked (Figure 4(b)), therefore this cell will be unblocked by calculating the $EucDist(S(5), Q(2))$. The value will be add to the SM which means that cell $SM(10)$ is now an entry in SM (Figure 4(c)). Although unblocking adds cells to SM which means the number of open cells will increase, but the overlapping in the bins bound-

Algorithm 3 *SparseDTW*: Sparse dynamic programming technique.

Input: S : Time series of length n , Q : Time series of length m , and Res .

Output: Optimal warping path and *SparseDTW* distance.

```

1:  $[S', Q'] \leftarrow \text{Quantize}(S, Q)$ 
2:  $LowerBound \leftarrow 0, UpperBound \leftarrow Res$ 
3: for all  $0 \leq LowerBound \leq 1 - \frac{Res}{2}$  do
4:    $IdxS \leftarrow \text{find}(LowerBound \leq S' \leq UpperBound)$ 
5:    $IdxQ \leftarrow \text{find}(LowerBound \leq Q' \leq UpperBound)$ 
6:    $LowerBound \leftarrow LowerBound + \frac{Res}{2}$ 
7:    $UpperBound \leftarrow LowerBound + Res$ 
8:   for all  $idx_i \in IdxS$  do
9:     for all  $idx_j \in IdxQ$  do
10:      Add  $\text{EucDist}(idx_i, idx_j)$  to  $SM$  {When  $\text{EucDist}(idx_i, idx_j) = 0, SM(i, j) = -1$ .}
11:    end for
12:  end for
13: end for
14: {Note:  $SM$  is linearly indexed.}
15: for all  $c \in SM$  do
16:    $LowerNeighbors \leftarrow \{(c-1), (c-n), (c-(n+1))\}$ 
17:    $minCost \leftarrow \min(SM(LowerNeighbors))$  { $SM(LowerNeighbors)=-1$  means  $cost=0$ .}
18:    $SM(c) \leftarrow SM(c) + minCost$ 
19:    $UpperNeighbors \leftarrow \{(c+1), (c+n), (c+n+1)\}$ 
20:   if  $|UpperNeighbors| == 0$  then
21:      $SM \cup \text{EucDist}(UpperNeighbors)$ 
22:   end if
23: end for
24:  $WarpingPath \leftarrow \Phi$ 
25:  $hop \leftarrow SM(n \times m)$  {Last index in  $SM$ .}
26:  $WarpingPath \cup hop$ 
27: while  $hop \neq SM(1)$  do
28:    $LowerNeighbors \leftarrow \{(hop-1), (hop-n), (hop-(n+1))\}$ 
29:    $[minCost, index] \leftarrow \min[Cost(LowerNeighbors)]$ 
30:    $hop \leftarrow index$ 
31:    $WarpingPath \cup hop$ 
32: end while
33:  $WarpingPath \cup SM(1)$ 
34: return  $WarpingPath, SM(n \times m)$ 

```

aries allows the SM 's unblocked cells to be connected mostly that means less number of unblocking operations. Figure 4(d) shows the final entries of the SM after calculating the warping cost of all open cells.

Lines 23 to 32 return the warping path. hop initially represents the linear index for the (m, n) entry of SM , that is the bottom right corner of SM in Figure 4(e). Starting from $hop = n \times m$ we choose the neighbors $[hop - n, hop - 1, hop - (n + 1)]$ with minimum warping cost and proceed recursively until we reach the first entry of SM , namely $SM(1)$ or $hop = 1$. It is interesting that while calculating the warping path we only have to look at the open cells, which may be fewer in number than 3. This potentially reduces the overall time complexity.

Figure 4(e) demonstrates an example of how the two time series (S and Q) are warped and the way their distance is calculated using *SparseDTW*. The filled cells show the optimal warping path, which crosses the grid from the top left corner to the bottom right corner. The distance between the two time series is calculated using Equation 4. Figure 4(f) shows the standard *DTW* where the filled cells are the optimal warping path. It is clear that both techniques

give the optimal warping path which will yield the optimal distance.

6.3 SparseDTW Complexity

Given two time series S and Q of length n and m , the space and time complexity of standard *DTW* is $O(nm)$. For *SparseDTW* we attain a reduction by a constant factor b , where b is the number of bins. This is similar to the *BandDTW* approach where the reduction in space complexity is governed by the size of the band. However, *SparseDTW* always yields the optimal alignment. The time complexity of *SparseDTW* is $O(nm)$ in the worst case as we potentially have to access every cell in the matrix.

7 Experiments, Results and Analysis

In this section we report and analyze the experiments that we have conducted to compare *SparseDTW* with other methods. Our main objective is to evaluate the space-time tradeoff between *SparseDTW*, *BandDTW* and *DTW*. We evaluate the effect of *correlation* on the running time of *SparseDTW*³. As we have noted before, both *SparseDTW* and *DTW* always yield the optimal alignment while *BandDTW* results can often lead to sub-optimal alignments, as the optimal warping path may lie outside the band. As we noted before *DC* may not yield the optimal result.

7.1 Experimental Setup

All experiments were carried out on a Windows XP operated PC with a Pentium(R) D (3.4 GHz) processor and 2 GB main memory. The data structures and algorithm were implemented in C++.

7.2 Datasets

We have used a combination of benchmark and synthetically generated datasets. The benchmark dataset is a subset from the *UCR* time series data mining archive (Keogh 2006). We have also generated synthetic time series data to control and test the effect of correlation on the running time of *SparseDTW*. We briefly describe the characteristics of each dataset used.

- **GunX:** comes from the video surveillance application and captures the shape of a gun draw with the gun in hand or just using the finger. The shape is captured using 150 time steps and there are a total of 100 sequences (Keogh 2006). We randomly selected two sequences and computed their similarity using the three methods.
- **Trace:** is a synthetic dataset generated to simulate instrumentation failures in a nuclear power plant (Roverso 2000). The dataset consists of 200 time series each of length 273.
- **Burst-Water:** is formed by combining two different datasets from two different applications. The average length of the series is 2200 points (Keogh 2006).
- **Sun-Spot:** is a large dataset that has been collected since 1818. We have used the daily sunspot numbers. More details about this dataset exists in (Vanderlinden 2008). The 1st column of the data is the year, month and day, the 2nd column is year and fraction of year (in

³The run time includes the time used for constructing the Sparse Matrix SM

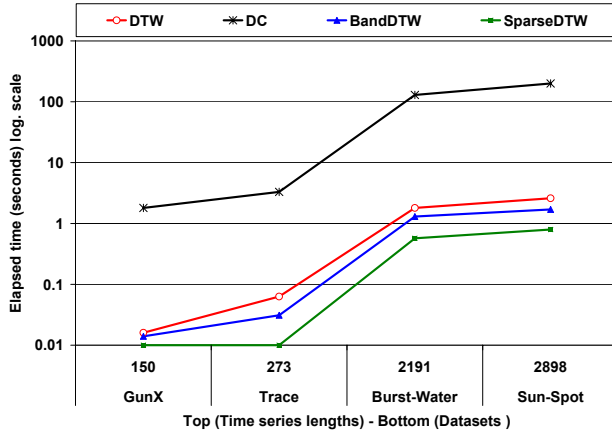


Figure 5: Elapsed time using real life datasets.

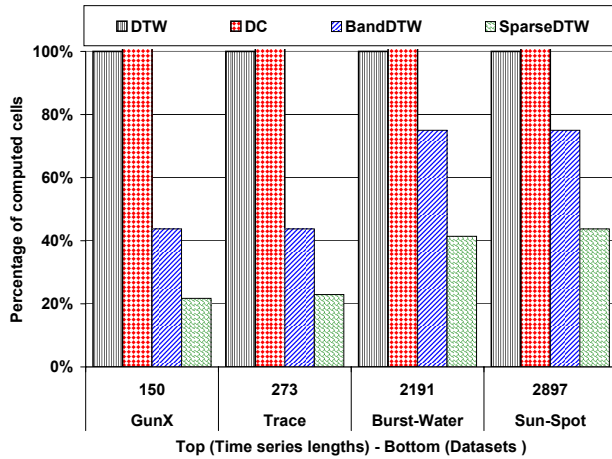


Figure 6: Percentage of computed cells as a measure for time complexity.

Julian year)⁴, and the 3rd column is the sunspot number. The length of the time series is 2898.

- **ERP**: is the Event Related Potentials that are calculated on human subjects⁵. The dataset consists of twenty sequences of length 256 (Makeig et al. 1999).
- **Synthetic**: Synthetic datasets were generated to control the correlation between sequences. The length of each sequence is 500.

Data size	Number of computed cells used by			
	DTW	DC	BandDTW	SparseDTW
2K	4×10^6	$> 8 \times 10^6$	2500	2000
4K	16×10^6	$> 30 \times 10^6$	5000	4000
6K	36×10^6	$> 70 \times 10^6$	7500	6000

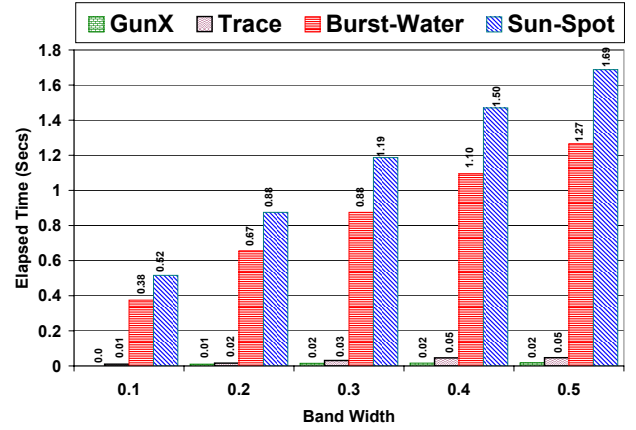
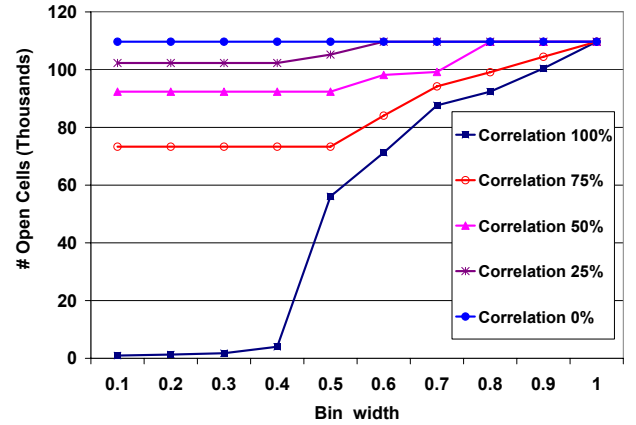
Table 2: Number of computed cells if the optimal path is close to the *diagonal*.

7.3 Discussion and Analysis

SparseDTW algorithm is evaluated against three other existing algorithms, *DTW*, which always gives the optimal answer, *DC*, and *BandDTW*.

⁴The Julian year is a time interval of exactly 365.25 days, used in astronomy.

⁵An indirect way of calculating the brain response time to certain stimuli

Figure 7: Effect of the band width on *BandDTW* elapsed time.Figure 8: Effects of the resolution and correlation on *SparseDTW*.

Dataset size	Algorithm name	#opened cells	Elapsed Time(Sec.)
3K	DTW	9×10^6	7.3
	SparseDTW	614654	0.65
6K	DTW	36×10^6	26
	SparseDTW	2048323	2.2
9K	DTW	81×10^6	N.A
	SparseDTW	4343504	4.8
12K	DTW	144×10^6	N.A
	SparseDTW	7455538	200

Table 3: Performance of the *DTW* and *SparseDTW* algorithms using large datasets.

Dataset name	Algorithm name	Number of opened cells	Warping path size (K)	Elapsed Time (Seconds)	DTW Distance
GunX	DTW	22500	201	0.016	0.01
	BandDTW	448	152	0.000	0.037
	SparseDTW	4804	201	0.000	0.01
Trace	DTW	75076	404	0.063	0.002
	BandDTW	1364	331	0.016	0.012
	SparseDTW	17220	404	0.000	0.002
Burst-Water	DTW	2190000	2190	1.578	0.102
	BandDTW	43576	2190	0.11	0.107
	SparseDTW	951150	2190	0.75	0.102
Sun-Spot	DTW	1266610	357	0.063	0.021
	BandDTW	12457	358	0.016	0.022
	SparseDTW	66049	357	0.016	0.021
ERP	DTW	1000000	1533	0.78	0.008
	BandDTW	19286	1397	0.047	0.013
	SparseDTW	210633	1535	0.18	0.008
Synthetic	DTW	250000	775	0.187	0.033
	BandDTW	4670	600	0.016	0.043
	SparseDTW	105701	775	0.094	0.033

Table 4: Statistics about the performance of *DTW*, *BandDTW*, and *SparseDTW*. Results in this table represent the average over all queries.

7.3.1 Elapsed Time

The running time of the four approaches is shown in Figure 5. The time profile of both *DTW* and *BandDTW* is similar and highlights the fact that *BandDTW* does not exploit the nature of the datasets. *DC* shows as well the worst performance due to the vast number of recursive calls to generate and solve sub-problems. In contrast, it appears that *SparseDTW* is exploiting the inherent similarity in the GunX and Trace data.

In Figure 6 we show the number of open/computed cells produced by the four algorithms. It is very clear that *SparseDTW* produces the lowest number of opened cells.

In Table 2 we show the number of computed cells that are used in finding the optimal alignment for three different datasets, where their optimal paths are close to the diagonal. *DC* has shown the highest number of computed cells followed by *DTW*. That is because both (*DC* and *DTW*) do not exploit the similarity in the data. *BandDTW* has shown interesting results here because the optimal alignment is close to the diagonal. However, *SparseDTW* still outperforms it.

Two conclusions are revealed from Figure 7. The first, the length of the time series affects the computing time, because the longer the time series the bigger the matrix. Second, band width influences CPU time when aligning pairs of time series. The wider the band the more cells are required to be opened.

DTW and *SparseDTW* are compared together using large datasets. Table 3 shows that *DTW* is not applicable (N.A) for datasets of size $> 6K$, since it exceeds the size of the memory when computing the warping matrix. In this experiment we excluded *BandDTW* and *DC* given that they provide no guarantee on the optimality.

To determine the effect of correlation on the elapsed time for *SparseDTW* we created several synthetic datasets with different correlations. The intuition being that two sequences with lower correlation will have a warping path which is further away from the diagonal and thus will require more open cells in the warping matrix. The results in Figure 8 confirm our intuition though only in the sense that extremely low correlation sequences have a higher number of open cells than extremely high correlation sequences.

7.3.2 SparseDTW Accuracy

The accuracy of the warping path distance of *BandDTW* and *SparseDTW* compared to standard *DTW* (which always gives the optimal result) is shown in Table 4. It is clear that the error rate of *BandDTW* varies from 30% to 500% while *SparseDTW* always gives the exact value. It should be noticed that there may be more than one optimal path of different sizes but they should give the same minimum cost (distance). For example, the size of the warping path for the *ERP* dataset produced by *DTW* is 1533, however, *SparseDTW* finds another path of size 1535 with the same distance as *DTW*.

Figure 9 shows the dramatic nature in which *SparseDTW* exploits the similarity inherent in the sequences and creates an adaptive band around the warping path. For both the GunX and the Trace data, *SparseDTW* only opens a fraction of the cells compared to both standard *DTW* and *BandDTW*.

8 Conclusions

In this paper we have introduced the *SparseDTW* algorithm, which is a sparse dynamic programming technique. It exploits the correlation between any two time series to find the optimal warping path between them. The algorithm finds the optimal path efficiently and accurately. *SparseDTW* always outperforms the algorithms *DTW*, *BandDTW* and *DC*. We have shown the efficiency of the proposed algorithm through comprehensive experiments using synthetic and real life datasets.

References

- Berndt, D. J. & Clifford, J. (1994), Using dynamic time warping to find patterns in time series, in 'Association for the Advancement of Artificial Intelligence, Workshop on Knowledge Discovery in Databases (AAAI)', pp. 229–248.
- Caiani, E., Porta, A., Baselli, G., Turie, M., Muzupappa, S., Piemzzi, Crema, C., Malliani, A. & Cerutti, S. (1998), 'Warped-average template technique to track on a cycle-by-cycle basis the cardiac filling phases on left ventricular volume', *Computers in Cardiology* **5**, 73–76.

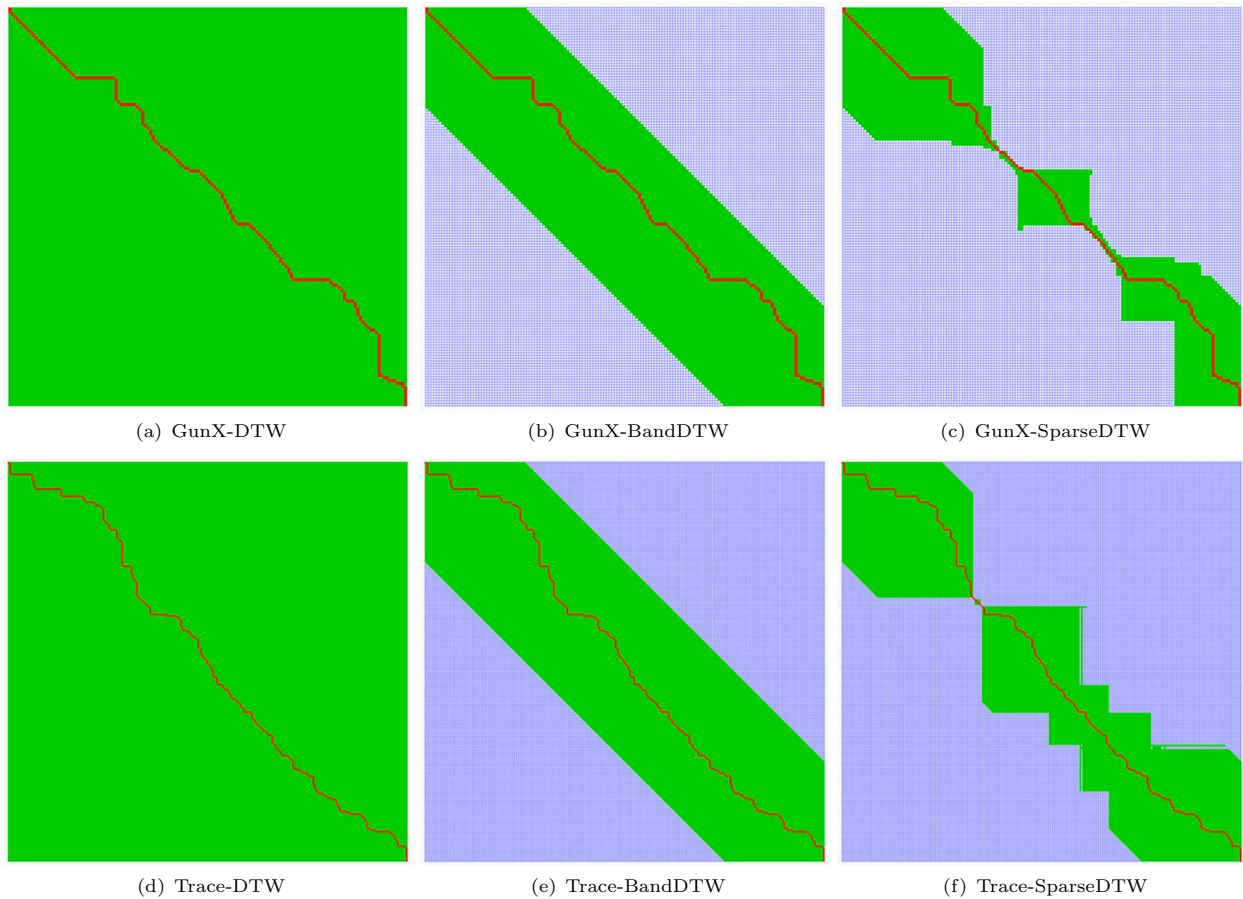


Figure 9: The optimal warping path for the GunX and Trace sequences using three algorithms (*DTW*, *Band-DTW*, and *SparseDTW*). The advantages of *SparseDTW* are clearly revealed as only a small fraction of the matrix cells have to be “opened” compared to the other two approaches.

- Capitani, P. & Ciaccia, P. (2007), ‘Warping the time on data streams’, *Data and Knowledge Engineering* **62**(3), 438–458.
- Hirschberg, D. (1975), ‘A linear space algorithm for computing maximal common subsequences’, *Communications of the ACM* **18**(6), 341–343.
- Itakura, F. (1975), ‘Minimum prediction residual principle applied to speech recognition’, *IEEE Transactions on Acoustics, Speech and Signal Processing* **23**(1), 67–72.
- Keogh, E. (2006), ‘The ucr time series data mining archive’, <http://www.cs.ucr.edu/~eamonn/TSDMA/index.html>.
- Keogh, E. & Pazzani, M. (2000), Scaling up dynamic time warping for datamining applications, in ‘Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)’, ACM Press, New York, NY, USA, pp. 285–289.
- Keogh, E. & Ratanamahatana, C. (2004), ‘Exact indexing of dynamic time warping’, *Knowledge and Information Systems (KIS)* **7**(3), 358–386.
- Kim, S.-W., Park, S. & Chu, W. (2001), An indexed approach for similarity search supporting time warping in large sequence databases, in ‘Proceedings of the 17th International Conference on Data Engineering (ICDE)’, IEEE Computer Society, Washington, DC, USA, pp. 607–614.
- Lemire, D. (2009), ‘Faster retrieval with a two-pass dynamic-time-warping lower bound’, *Pattern Recogn.* **42**(9), 2169–2180.
- Makeig, S., Westerfield, M., Townsend, J., Jung, T.-P., Courchesne, E. & Sejnowski, T. (1999), ‘Functionally independent components of early event-related potentials in a visual spatial attention task’, *Philosophical Transaction of The Royal Society: Biological Science* **354**(1387), 1135–1144.
- Myers, C., Rabiner, L. R. & Rosenberg, A. E. (1980), ‘Performance tradeoffs in dynamic time warping algorithms for isolated word recognition’, *IEEE Transactions on Acoustics, Speech and Signal Processing* **28**(6), 623 – 635.
- Rabiner, L. & Juang, B.-H. (1993), *Fundamentals of speech recognition*, Prentice Hall Signal Processing Series, Upper Saddle River, NJ, USA.
- Roverso, D. (2000), Multivariate temporal classification by windowed wavelet decomposition and recurrent neural networks, in ‘Proceedings of the 3rd ANS International Topical Meeting on Nuclear Plant Instrumentation, Control and Human-Machine Interface Technologies (NPIC and HMIT)’.
- Sakoe, H. & Chiba, S. (1978), ‘Dynamic programming algorithm optimization for spoken word recognition’, *IEEE Transactions on Acoustics, Speech and Signal Processing* **26**(1), 43– 49.
- Sakurai, Y., Yoshikawa, M. & Faloutsos, C. (2005), FTW: Fast similarity search under the time warp-

ing distance, *in* 'Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (PODS)', ACM, New York, NY, USA, pp. 326–337.

Salvador, S. & Chan, P. (2007), 'Toward accurate dynamic time warping in linear time and space', *Intelligent Data Analysis* **11**(5), 561 – 580.

Schmill, M., Oates, T. & Cohen, P. (1999), Learned models for continuous planning, *in* 'The Seventh International Workshop on Artificial Intelligence and Statistics (AISTATS)', pp. 278–282.

Tappert, C. C. & Das, S. K. (1978), 'Memory and time improvements in a dynamic programming algorithm for matching speech patterns', *IEEE Transactions on Acoustics, Speech and Signal Processing* **26**(6), 583– 586.

Vanderlinden, R. (2008), 'Sunspot data', <http://sidc.oma.be/html/sunspot.html>.
URL: <http://sidc.oma.be/html/sunspot.html>

Yi, B.-K., Jagadish, H. V. & Faloutsos, C. (1998), Efficient retrieval of similar time sequences under time warping, *in* 'Proceedings of the Fourteenth International Conference on Data Engineering (ICDE)', IEEE Computer Society, Washington, DC, USA, pp. 201–208.

HDAX: Historical Symbolic Modelling of Delay Time Series in a Communications Network

Hooman Homayounfar

Paul J. Kennedy

Centre for Quantum Computation and Intelligent Systems,
Faculty of Engineering and Information Technology,
University of Technology, Sydney,
PO Box 123, Broadway NSW 2007, AUSTRALIA,
Email: hoomanhm@it.uts.edu.au, paulk@it.uts.edu.au

Abstract

There are certain performance parameters like packet delay, delay variation (jitter) and loss, which are decision factors for online quality of service (QoS) traffic routing. Although considerable efforts have been placed on the Internet to assure QoS, the dominant TCP/IP - like the best-effort communications policy - does not provide sufficient guarantee without abrupt change in the protocols. Estimation and forecasting end-to-end delay and its variations are essential tasks in network routing management for detecting anomalies. A large amount of research has been done to provide foreknowledge of network anomalies by characterizing and forecasting delay with numerical forecasting methods. However, the methods are time consuming and not efficient for real-time application when dealing with large online datasets. Application is more difficult when the data is missing or not available during online forecasting. Moreover, the time cost in statistical methods for trivial forecasting accuracy is prohibitive. Consequently, many researchers suggest a transition from computing with numbers to the manipulation of perceptions in the form of fuzzy linguistic variables. The current work addresses the issue of defining a delay approximation model for packet switching in communications networks. In particular, we focus on decision-making for smart routing management, which is based on the knowledge provided by data mining (informed) agents. We propose a historical symbolic delay approximation model (HDAX) for delay forecasting. Preliminary experiments with the model show good accuracy in forecasting the delay time-series as well as a reduction in the time cost of the forecasting method. HDAX compares favourably with the competing Autoregressive Moving Average (ARMA) algorithm in terms of execution time and accuracy.

Keywords: Time Series Data Mining, Piecewise Linear Approximation, Perception-based Approximation, Delay Forecasting.

1 Introduction

Rapid increases in the number of Internet users and services have prompted researchers within academia and industry to contemplate smart ways of supporting applications with the required Quality of Service (QoS) (Rankin et al. 2005). QoS is the measure of

transmission quality and service availability of a network (or internetworks). Service availability is a crucial part of QoS and the network infrastructure must be designed so as to provide high availability to meet QoS. The target of 99.999% availability permits five minutes of downtime per year. Some important metrics for measurement of QoS in the network are:

Delay – The finite amount of time it takes a packet for an end-to-end transmission from sender to receiver. These (sender and receiver) are both edge routers in the same autonomous system in our case.

Delay Variation (Jitter) – The difference in the end-to-end delay between packets. For instance, if one packet and the following packet requires 125 ms and 100 ms, respectively, to traverse their end-to-end trip, then the delay variation would be 25 ms.

Packet Loss – A relative measure of the number of data packets lost during transmission compared to the total number of packets transmitted. Loss is typically a function of availability. That is, if the network is highly available, then loss during periods of non-congestion is zero. However, QoS decision-makers prioritise packets to be transmitted to a link to reduce the likelihood of loss for a specific service level agreement.

By prioritising Internet traffic and the core network more efficiently, QoS and traffic engineering functions can address performance issues related to emerging Internet applications such as real-time voice and video streaming. Consequently, technologies such as those based on software agents¹ are expected to become key tools for the development of future software (Debenham & Simoff 2007). This type of technology may be used in distributed telecommunication environments such as mobile computing, e-commerce and routing management. An effective routing mechanism and its management is crucial to satisfactorily support diverse services in such networks. Routing tables, as the maps in packet delivery throughout the network, are dynamic and get updated by network state-based events (Lau & Woo 2007). Typical network events include node failure, link failure and congestion. A major problem with current routing mechanisms is that they generate routing tables that do not reflect the real-time state of the network and ignore factors like local congestion.

Although considerable efforts have been placed on the Internet to assure QoS, the dominant TCP/IP - like the best-effort traffic handling policy - does not provide sufficient QoS guarantee without abrupt change in the protocols (Bui et al. 2007). Estimation, approximation and forecast of delay variations are essential tasks in network routing management for detecting anomalies. A large amount of research has been done to provide foreknowledge of the

Copyright ©2009, Australian Computer Society, Inc. This paper appeared at the Eighth Australasian Data Mining Conference (AusDM 2009), Melbourne, Australia. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 101, Paul J. Kennedy, Kok-Leong Ong and Peter Christen, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

¹The term *agent* is employed throughout this paper as the automated tool (software) which is capable of acting independently in the network without human monitoring.

network anomalies by characterizing and forecasting delay variation through using numerical forecasting methods. However, the methods are time consuming and/or not efficient for real-time application whilst we are dealing with online and large datasets particularly when data is missing or unavailable. Moreover, the time cost in statistical methods for trivial forecasting accuracy is prohibitive. Zadeh (2001), Keogh et al. (2005), and Batyrshin & Sheremetov (2008) suggest a transition from computing with numbers to the manipulation of perceptions in the form of linguistic variables defined in fuzzy logic.

According to Zadeh (2001), what Lord Kelvin, Rudolph Kalman, William Kahan, and many other outstanding minds did not acknowledge is the fundamental human ability of performing physical and mental works without any complicated numerical computations. Common instances of these works are parking a car; driving in heavy traffic; playing guitar; understanding speech, and summarizing a journal paper. In all cases, the brain works with fuzzy perceptions – perceptions of size, distance, weight, speed, time, direction, smell, colour, shape, force, likelihood, truth and intent, among others – provided by five human senses instead of running a complicated algorithm in the background. In our especial case of delay forecasting model, we aim to simulate this remarkable ability of the brain by memorising the patterns' frequency in the delay (jitter) time series within sliding windows of time. We may later use the provided foreknowledge and manipulate perceptions of delay trends and current delay value to forecast the next trends and values – instead of running a statistical algorithm each time.

The majority of packet loss in the Internet occurs due to congestion in the links. Real-time multimedia applications over the Internet such as Voice over IP (VOIP), video/audio conferencing and streaming are sensitive to packet loss, and packet retransmission is not an acceptable solution with these sorts of application. Predicting packet loss with some certainty from network dynamics of past transmissions is crucial knowledge to inform smart routers to make better decisions. Network applications such as wireless need to identify congestion in the path of packets so as to take suitable rate control actions (Roychoudhuri & Al-Shaer 2005). We propose a data mining model for classification of links that have a high probability of packet loss. The model is intended to contribute to making informed decisions within smart edge routers where the quality of transmissions should be controlled and is primarily determined by the level packet loss.

The basic idea of our proposed model is to predict packet loss in a link by approximating the trends and values of the delay according to observed patterns. We propose a framework utilised by intelligent software agents for data mining, which are installed on edge routers. The agent predicts the likelihood of packet loss for the link, according to a predicted delay level, and periodically broadcasts the availability of the link to its neighbours. This will prevent other links from sending packets to links that report congestions within their network autonomous system. The task becomes intractable when there is no real-time data to accomplish online data mining. In this paper we describe our framework that approximates the probability of packet loss based on the predicted delay values.

The rest of this paper is organised as follows. Section 2 has two parts, the first describing related work in predicting performance traces focusing on delay forecasting and the next describing different network data sources, including the one used in this paper. In section 3 we explain our approach to forecasting net-

work delay values and in section 4 we present results of our experiments on applying our model. Section 5 concludes the paper.

2 Background

2.1 Related Work

Internet packet-loss and delay variation show temporal dependency. If packet n is lost, packet $n + 1$ is likely to be lost. This leads the network to a “bursty” packet loss in real-time communications network (Jiang & Schulzrinne 2000) that may degrade quality of service and the effectiveness of recovery mechanisms such as Forward Error Correction (FEC). In this regard, various researches have studied the delay and packet loss, jitter, available bandwidth and other performance traces in the network so as to predict network anomalies or differentiate them from noise.

A quantitative study of delay and loss correlation patterns from off-line analysis of measurement data from the Internet has been done by Moon (2000), although it did not consider real-time prediction of packet loss from the delay variation data of online communications. A correlation between the bandwidth used and the amount of observed loss is also reported by the researchers while measuring performance of their Mbone system (Handley 1997, Hermanns & Schuba 1996). The path-load measurement in Jain & Dovrolis (2002) employs delay variations to calculate the available bandwidth. However, as confirmed in Roychoudhuri & Al-Shaer (2005), there is not a detailed analysis of online prediction of the packet loss from the delay variation.

Some researchers report techniques using delay variation and TCP window size for congestion avoidance (Brakmo et al. 1994, Jain 1989, Wang & Crowcroft 1991). Others use these results for classification of loss and noise in wireless transmission (Biaz & Vaidya 1998). Moreover, Tobe et al. (2000) discriminates various degrees of congestion according to the relative *one way delay* and *static delay thresholds*. They did not measure or predict delay trends and packet loss based on the congestion level (Roychoudhuri & Al-Shaer 2005).

Other research projects (Carter & Crovella 1996, Dovrolis et al. 2001, Paxson 1998) have employed packet routing techniques to estimate the path capacity and available bandwidth. Specifically, Paxson (1998) examined the correlation between delay and packet loss, but concluded that the linkage between delay variation and loss was weak, though not negligible. In contrast with Paxson's observations, Roychoudhuri & Al-Shaer (2005) attempted to predict packet loss based on delay variation patterns, rather than the overall amount of delay variation. They developed a technique to detect congestion and predict packet loss by means of inter-packet gap variations observations.

Our work addresses the issue of defining a symbolic delay forecasting model, which is based on historical frequencies of observed patterns of the delay variation in adjacent TCP windows. In particular, we will focus our research to produce an “informed” (Debenham & Simoff 2007, Rocha-Mier et al. 2007, Miloucheva et al. 2003) data mining model to be used in a smart network router for online routing management.

2.2 Data Network Scenarios

It is obvious that the type of data used in data mining is highly dependent on the case studies and research goals. For instance, customer data cannot be used in

lieu of network data for predicting faults in a network. However, when making real-time informed decisions before partial network congestion, use of other data repositories may be useful for the decision-making (Weiss et al. 1998). Weiss et al. (1998) also list three types of data used in telecommunications networks: call data, network data and customer data. However, the type of data needed for routing management is network data and there are two salient types mentioned in the literature for fault prediction and isolation: alarm data and QoS data traces.

Rocha-Mier et al. (2007) describe measurement and modelling sequences of various network variables that comprise of time series based on data network statistics. They have created a useful network scenario using OPNET Modeller. The modeller was employed for generating simulated statistics and values of network data variables. Although real network data variables could be derived from the data logs by the use of intelligent agents or manually by the system administrators, there may be violation in accessing data throughout the Internet. Therefore, they adopted the modeller to study various levels of the network traffic load as well as types of services and applications.

Network sequences derived from alarm databases have been used in Klemettinen (1999). Telecommunications databases contain event data from a number of interconnected components, such as switches. Traces used by Klemettinen were produced by the components to report abnormal situations. Klemettinen (1999) views an occurrence a of an alarm as a triple $a = (t, s, m)$, where t is the time of the alarm, s is the “sender” of the alarm, and m is the “alarm message”. The alarm “episodes” have all the information about the problems (Weiss et al. 1998). Alarm data are also used by Hatonen et al. (1996) in their knowledge discovery system called telecommunications alarm sequence analyser.

Telecommunications network QoS data traces, on the other hand, are used in novel data mining methods for automation of pattern recognition. These data mining methods use similarity analysis and management of significant patterns in network performance datasets (Miloucheva et al. 2003). This involves analysing the time series of QoS data traces and is required for boosting performance of QoS data mining in network autonomous fault prediction and isolation.

In our experiments, we used delay traces in TCP-dump format from network traffic archives generated by Napoli University “Federico II”. The data traces were simulated from real network topologies using Distributed Internet Traffic Generator (D-ITG) (Botta et al. 2007). D-ITG is a software platform that generates traffic at network, transport and application layers over Linux and Windows platforms. It is capable of producing IPV4 and IPV6 traffic traces at packet level accuracy, and replicating appropriate stochastic processes for both Inter Departure Time and Packet Size random variables. The random variables may be obtained from a variety of probability distributions such as exponential, uniform, Cauchy, Gaussian and Pareto.

The generated archives are kept in tar.gz format, each of which contains sample datasets of QoS parameters - packet loss, delay and jitter. The archives are related to several end-to-end paths. Botta et al. (2007) report that samples are generated by adopting an active measurement approach and sending probe packets of 64, 512 and 1024 bytes with a packet rate of 100 packets per second. For each generation experiment for production of One way Delay, Round Trip Time, packet loss and jitter traces it is possible to initialise the random variables seed.

We use non-stationary stochastic delay time series of TCP, UDP and SCTP probe packets sent at regular time intervals to characterise the end-to-end packet delay behaviour of the Internet. The time series are ON/OFF background traffic sources, calculated using non-overlapping windows of 10ms length, for wired, wireless and ADSL network. The data are described in more detail in section 4.

3 Proposed Forecasting Algorithm: Historical Symbolic Delay Approximation (HDAX)

In this section we describe our approach to forecasting the delay values from previously observed patterns of the time series of delay values. Our approach consists of two phases: training and simulation. The training phase uses a time series dataset of delay values to recognise patterns and make the look-up pattern matrix (PDB) mapping the trend patterns to their observed frequency. This phase moves a sliding window over the training data and makes patterns consisting of 3 consecutive trends (previous, current and next) together with their frequency. PDB is then used in the simulation phase on real data to predict the next trend and associated delay value from the current and previously observed trend patterns.

The patterns are described in terms of linguistic variables as a perception-based function (PBF). According to Zadeh (2001) and Batyrshin & Sheremetov (2008), a perception-based function is a fuzzy function resulting from human perception and is given by the rule sets of R_i : “if X is T_i then Z is S_i ” where T_i is a categorical (linguistic) term describing some fuzzy intervals A_i on the domain of real numbers and S_i are the descriptions of the shape of $Y(x)$ function on these intervals. For example, regarding the network expert empirical knowledge about delay trends in the network, the PBF rules set is defined as

$$R: \text{“if } X \text{ is } T_k \text{ then } Y_k \text{ is } S” , (k = 1, \dots, l) \quad (1)$$

where Y_k is the trend of the delay time series in the time window k (intervals with 10 seconds length each) and S is the perception-based categorical value of the delay PBF trend defined as *Plain*, *Increase*, *Sharply Increase*, *Decrease*, *Sharply Decrease* and *Outliers*.

In our case, we represent possible future trends of the QoS time series y_t of delay values at time intervals t and $t - 1$ with categorical terms. Note that we use y_t to denote the delay value at time t and Y_k to denote the trend (linguistic variable) for window k . For simplicity, the linearly ordered scale in our experiment also has six linguistic grades (defined above) each of which is a categorical term (assigned to the case number of zero to five respectively).

$$\text{Alphabet} = \langle SI, I, P, D, SD, OUT \rangle \quad (2)$$

The interpretation of these categorical grades is de-

Table 1: The scale of delay trends (PBF).

Case	Id	Description	$Y = y_t - y_{t-1}$
0	P	Plain	$Y = 0$
1	I	Increase	$0 < Y \leq \max/2$
2	SI	Sharply Increase	$\max/2 < Y \leq \max$
3	D	Decrease	$-\max/2 \leq Y < 0$
4	SD	Sharply Decrease	$-\max \leq Y < -\max/2$
5	OUT	Outlier	$Y < -\max \text{ or } Y > \max$

scribed in Table 1 which relates the case number, the

abbreviation, description of the categorical abbreviation and its meaning in terms of the change in values between two consecutive values of the time series Y .

Another notion in our algorithm is the delay level function, which shows the level of delay observed at time t . This value is needed because we might see two kinds of pattern frequency, but in different contexts, as shown in the Fig. 1.

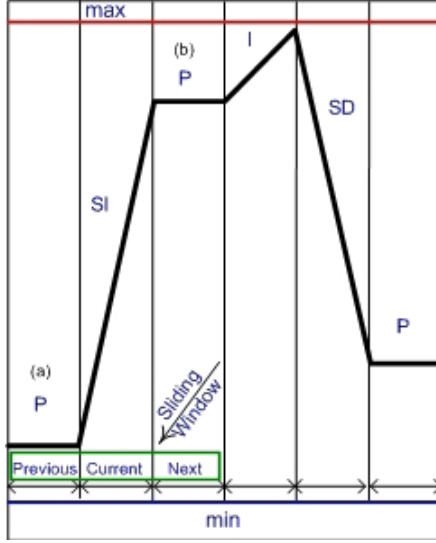


Figure 1: A sample string of symbolic values for the average delay variation time series $\{P, SI, P, I, SD, P\}$

Figure 1 shows that we may have two (or more) contexts of having “plain” trend (labelled as P) in slope sequences. For instance, we may have two cases: a) when the value of y_t is near the minimum and b) when the value of y_t is near the maximum. Therefore, we also need to store the delay level value (ie. whether it is close to the minimum or maximum value) so as to have a finer classification between these two different cases. Consequently, we define a delay level function $F(Y_t)$ as

$$F(Y_t) = \begin{cases} 0 & \text{if } 0 \leq Y_t < \max/4 \\ 1 & \text{if } \max/4 \leq Y_t < 2 \times \max/4 \\ 2 & \text{if } 2 \times \max/4 \leq Y_t < 3 \times \max/4 \\ 3 & \text{if } 3 \times \max/4 \leq Y_t \leq \max \\ 4 & \text{if } \max \leq Y_t \end{cases} \quad (3)$$

The \max parameter used in Table 1 and equation (3) is defined by the user and in this work was empirically set to 60000 ms based on the training dataset used.

After defining the delay level function, we introduce the pattern database (PDB) which is a matrix with a row for each pattern and 6 columns as listed in Table 2. The PDB relates a triplet of values for trends (as listed in Table 1) for the previously observed trend, the current trend and the next trend together with the frequency of the pattern.

Next we describe the training and simulation phases in more detail.

3.1 Training Phase

In this phase, we slide three adjacent windows (previous, current and next) each of length 10 along the training data and increment the frequency value for the associated pattern in the PDB. The window length of 10 was chosen empirically as a balance between performance of the algorithm and generalisation of trends. If the value is too high, there may

Table 2: Description of the fields in the Pattern Database

Field Name	Field Type	Description
Index	Integer	Index of the table
Prev	Categorical $\{0, 1, 2, 3, 4, 5\}$	Categorical based on the value of $y_t - y_{t-1}$
Current	Categorical $\{0, 1, 2, 3, 4, 5\}$	Categorical based on the value of $y_t - y_{t-1}$
Next	Categorical $\{0, 1, 2, 3, 4, 5\}$	Categorical based on the value of $y_t - y_{t-1}$
fyt	Categorical $\{0, 1, 2, 3, 4\}$	Categorical based on the $F(Y_t)$ value
Frequency	Long integer	N/A

be too long a time to detect anomalies in delay values. If the value is too low, too much overhead may be placed on the router. The 10 values in a window are averaged to calculate the y_t values, and y_{t-1} and y_t are used to calculate the trend using the scheme outlined above and in Table 1.

For simplicity, we implemented the PDB as a MATLAB matrix, but alternatively it may be created as a physical table in a database. Our PDB matrix contained 1080 rows and 6 columns, as we have an alphabet containing 6 symbols (P, I, SI, D, SD, OUT) and 5 Y_t level ($6 \times 6 \times 6 \times 5 = 1080$).

3.2 Simulation Phase

Once the PDB is populated by training it over a sufficiently long sequence of delay values, it can be used to predict delay trends and values for a new sequence of delay values. We call this the simulation phase.

In this phase we observe the previous 2 delay average values, grouped into two adjacent windows (previous and current patterns) each of length 10, to forecast the next average delay value and trend. The trends for the previous and current windows are calculated in the same way as described in the previous section. The two consecutive (previous and current) patterns (Y_{t-1} and Y_t) are looked up in the PDB and the associated “next” pattern with the highest frequency is chosen as the expected next trend (ie. Y_{t+1}). The next delay value y_{t+1} is estimated from Y_{t+1} (ie. the “next” trend from the PDB) and the “current” delay value y_t using the $F(Y_{t+1})$ function defined as

$$F(Y_{t+1}) = \begin{cases} y_t & \text{if } Y_{t+1} = 0 \\ y_t + \max/4 - \sigma & \text{if } Y_{t+1} = 1 \\ y_t + 3 \times \max/4 - \sigma & \text{if } Y_{t+1} = 2 \\ y_t - \max/4 + \sigma & \text{if } Y_{t+1} = 3 \\ y_t - 3 \times \max/4 + \sigma & \text{if } Y_{t+1} = 4 \\ y_t \pm \max \pm \sigma & \text{if } Y_{t+1} = 5 \end{cases} \quad (4)$$

where y_t is the current delay value, \max is a threshold for detecting the outliers and maximum delay without packet loss and σ is the standard deviation of the y_t values in the training set. The \max value was set as in Table 2. In the last condition of equation (4) the \max and σ values are added if $y_t > \max$, otherwise they are subtracted.

3.3 Measuring the quality of HDAX

An alternative approximation method defined in Box et al. (2008) to approximating the time series value is the Autoregressive Moving Average. An abbreviation of $ARMA(p, q)$ is written in different literature for the mixed autoregressive moving average model with p autoregressive and q moving average terms. As shown in Chatfield (2001), given a time series of data X_t where t is an integer index (indicating time intervals in the experiments) and the X_t are real numbers, then an $ARMA(p, q)$ model is written as

$$(1 - \sum_{i=1}^p \phi_i)X_t = (1 + \sum_{i=1}^q \theta_i)\epsilon_{t-1} \quad (5)$$

where p and q are the orders and the ϕ_i and θ_i are the coefficients of the autoregressive and the moving average parts, respectively. The ϵ_t are residuals in prediction, defined by the following recurrent expressions (\hat{Y}_t and Y_t are respective forecasted and original delay values at time t):

$$\epsilon_t = |\hat{Y}_t - Y_t| \quad (6)$$

The version of ARMA used here has four arguments as its input and generates one step ahead of the entered ARMA time series. The four arguments are the time series value X_t , the ARMA coefficients ϕ and θ , and a control argument *yesmean* that takes one of the two values 1 and 0 – whether the mean is subtracted from the input time series or not.

Since the ARMA model is autoregressive we have to compute the impulse response, which are the coefficients of the equivalent moving average representation. The p and q orders, of ARMA sequence, X_t , may be estimated by minimising the criterion reported in Hannan & Rissanen (1982). However, in our experiments, we set the p and q orders of the ARMA time series empirically. Since \hat{Y}_t is a function of ϕ and θ we estimate ϕ and θ by minimising the logarithm of the sum of squared errors between the actual and predicted values over a window w

$$\log \sum_{t=1}^w \epsilon_t^2 \quad (7)$$

where ϵ_t is the forecasting error at time t . Based on this, the optimum values of ARMA coefficients be set to $\phi = (1, -0.6)$ and $\theta = (0.6^0, 0.6^1, \dots, 0.6^j, \dots, 0.6^{10})$.

To compare the HDAX predicted delay values with the known delay values (from test data) we define a distance function to measure their similarity. For two time sequences $S = s_1, \dots, s_n$ and $Q = q_1, \dots, q_n$ with the same length of n , the Euclidean distance of S and Q is

$$D(S, Q) = \sqrt{\sum_{i=1}^n (s_i - q_i)^2} \quad (8)$$

In our experiments, we do need to calculate a distance function so as to estimate the accuracy of the algorithms. The HDAX and ARMA results may be compared to test data based on the number of forecasting steps, window sizes, elapsed time and the normalized root mean squared error (NRMSE). This latter is defined as

$$NRMSE = \sqrt{\frac{\sum_{t=1}^l (y_t - \bar{y}_t)^2}{\sum_{t=1}^l y_t^2}} \quad (9)$$

where y_t and \bar{y}_t are delays and forecasted delays, respectively, and $l < n$ is the length of the subsequence of the original time series. To make the comparison more efficient, we used a comparable, but less complex, error defined as

$$NRMSE = \sqrt{\frac{\sum_{t=1}^l (y_t - \bar{y}_t)^2}{n^2 \times (\max(y_t) - \min(y_t))^2}} \quad (10)$$

4 Simulation Results

In this section we present simulation results for HDAX and ARMA in terms of the accuracy and speed of the algorithm. The experiments ran on a Compaq nx7010 laptop (CPU: Intel Pentium M 1.70GHz; RAM: 1280MB) and a server cluster node with 2x 3.33 Ghz 8MB cache Quad Core Xeon W5590 6.4GT/sec QPI, 24GB(6x4GB) DDR3-1333 ECC Memory and 2 x 300GB 15,000 RPM SAS Hard Drive (Raid 1).

In terms of data for the preliminary experiments with HDAX and ARMA, we used archives that had been generated by Napoli University “Federico II” using the I-DTG platform. As introduced before, the generated archives were ON/OFF background TCP traffic sources for wired, wireless and ADSL network with 100Kbps transfer rate, 500ms burst time and 500ms idle time. Each archive contains a time series of delay values with packet sizes of 64, 256, 512 and 1024 bytes. Each sample is calculated using non-overlapping windows of 10ms length. The non-stationary traces are collected with no time synchronization between sender and receiver computers. Thus, the delay samples can not be used to estimate the average delay, but rather the delay distribution and variance.

In the first simulation phase, to provide more visibility for comparing the forecasting results of both HDAX and ARMA, we used traces of one wired network traffic archive with the packet size of 64 bytes in TCP-dump format. To start using the data for HDAX and ARMA simulations, the original dataset was transformed (3160 delay values) into 316 windows of equal length 10 and the average values of each segment computed. These average values are used in HDAX as described above. The window length of 10, used in these experiments is not suitable for all real-world network scenarios. A network expert should make a trade-off between the algorithm overhead and the performance of the anomaly detection to define the proper window length.

We divided the 316 average delay values into two datasets for training and simulation phases. The first 30% of the data is used in the training phase to build and populate the PDB as described in section 3.1. This comprised 100 y_t delay values. The remaining 70% of the data (216 y_t values) was used for 4 continuous simulation runs each with 54 values. We ran HDAX simulation phase for each set of 54 values to forecast delay values. We then calculated the overall NRMSE using equation (10). Predicted and original delay values for each of the four HDAX and ARMA simulation runs in simulation phase 1 are shown in Figures 2 to 5. These results show that the predicted delay values are close to the expected values in most cases. Figures 2 to 4 show that HDAX sometimes predicts a higher delay than expected when there are sharp increases. Similarly, figures 3 and 4 show lower than expected delay values when there are sharp decreases in delay value. Future work will examine the causes for these misjudgements.

Table 3 shows the normalised root mean square errors (NRMSE) for the four simulation runs in sim-

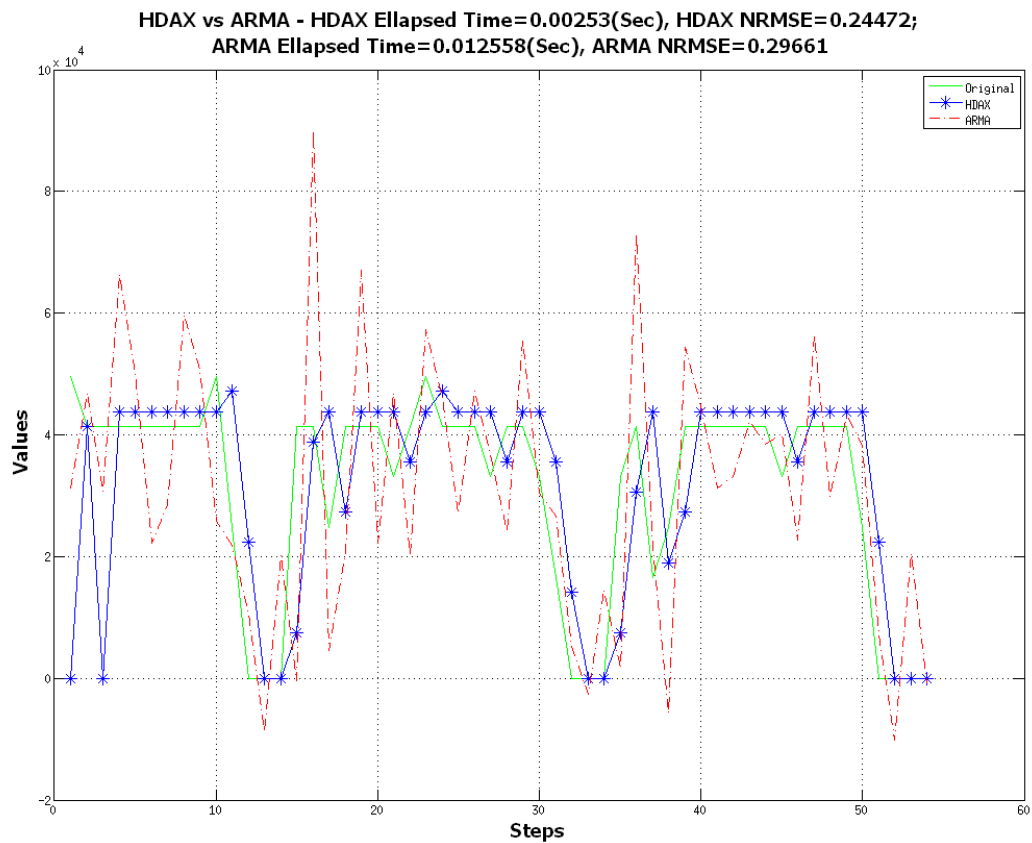


Figure 2: Original (solid line), HDAX predicted (star-dashed line) and ARMA predicted (dot-dashed line) delay values for simulation phase 1 run 1.

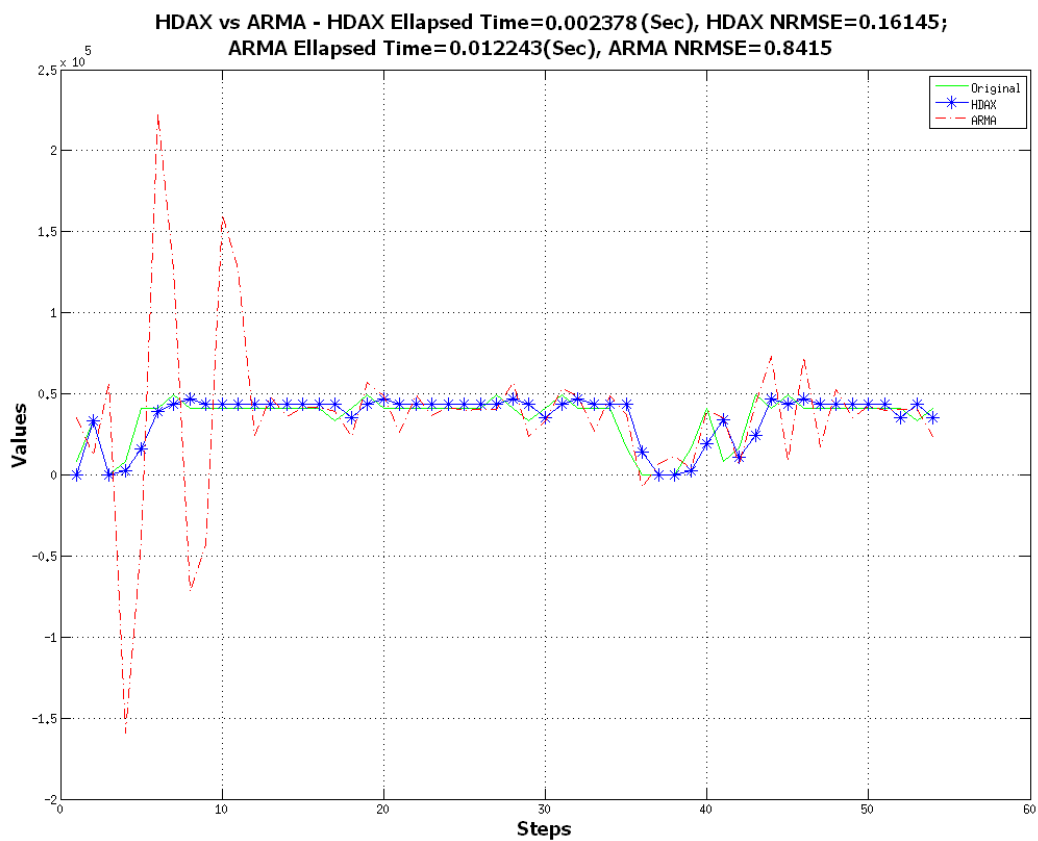


Figure 3: Original (solid line), HDAX predicted (star-dashed line) and ARMA predicted (dot-dashed line) delay values for simulation phase 1 run 2.

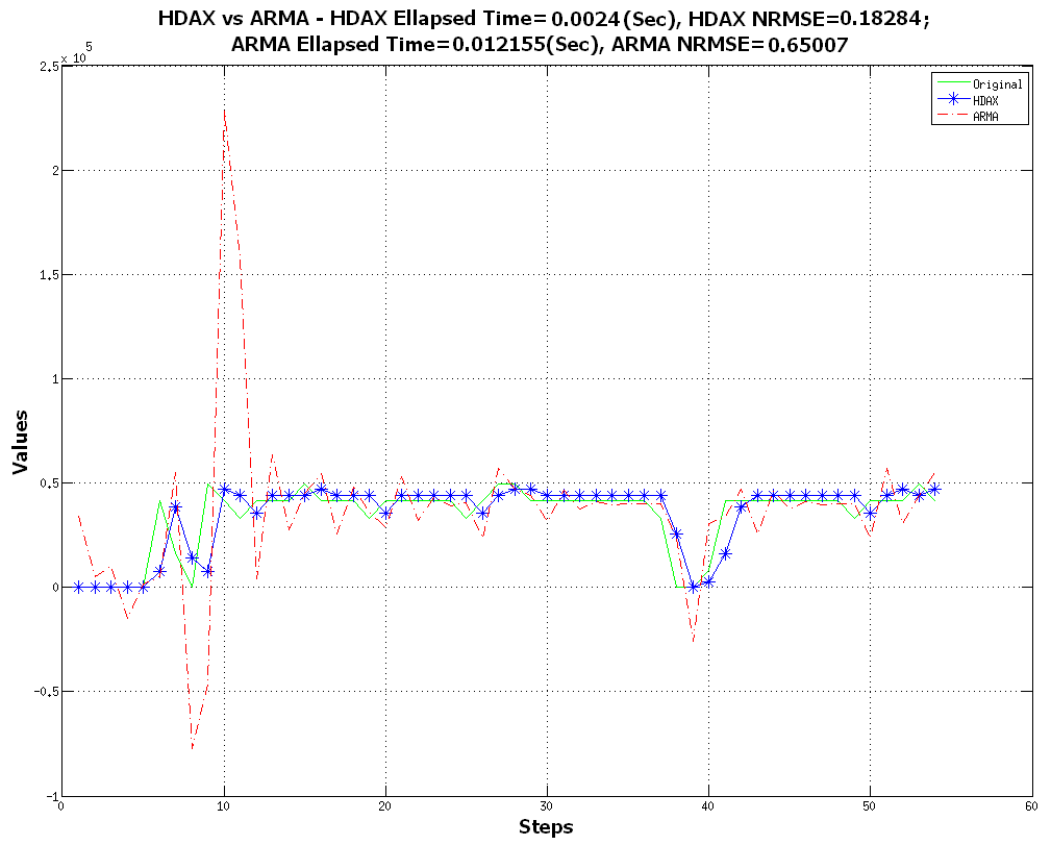


Figure 4: Original (solid line), HDAX predicted (star-dashed line) and ARMA predicted (dot-dashed line) delay values for simulation phase 1 run 3.

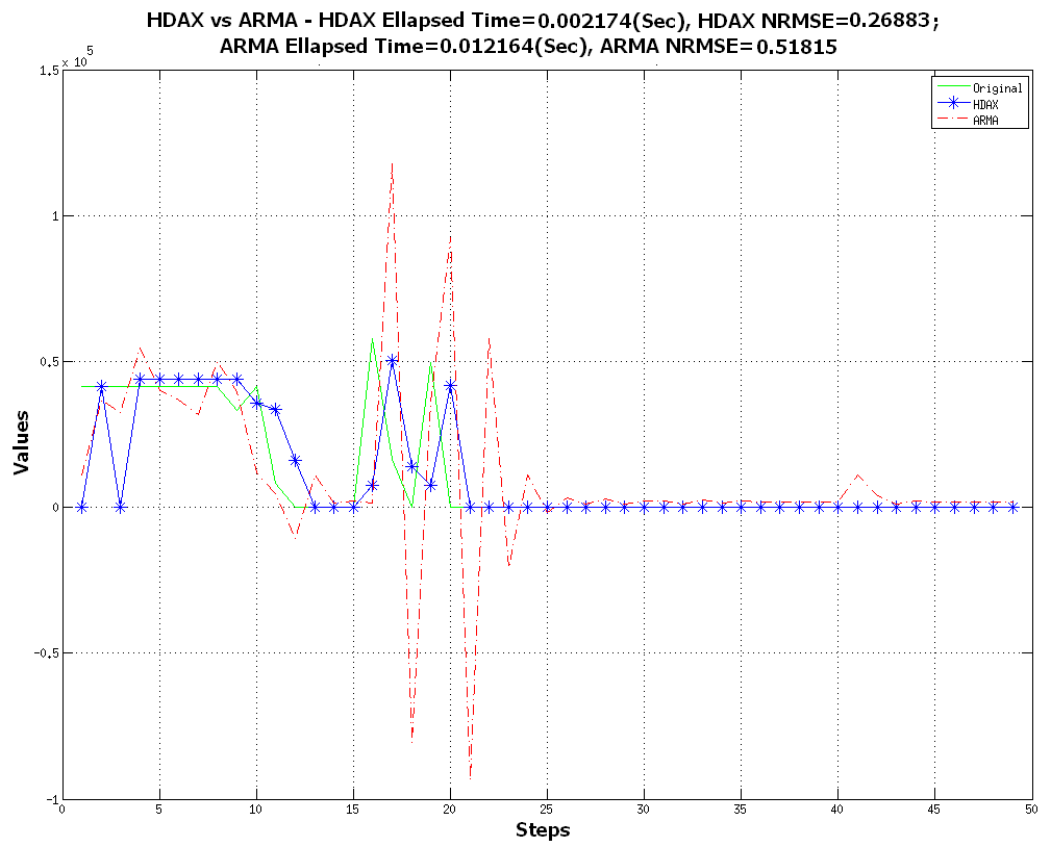


Figure 5: Original (solid line), HDAX predicted (star-dashed line) and ARMA predicted (dot-dashed line) delay values for simulation phase 1 run 4.

ulation phase 1 experiments together with their speed of calculation.

Table 3: Accuracy of HDAX and ARMA (benchmark) on first phase of simulation runs together with speed of calculation.

Simul. Run	HDAX NRMSE	Speed (sec)	ARMA NRMSE	Speed (sec)
1	0.24472	0.00253	0.29661	0.012558
2	0.16145	0.002378	0.8415	0.012243
3	0.18284	0.0024	0.65007	0.012155
4	0.26883	0.002174	0.51815	0.012164

In the second phase of simulations we ran the HDAX and ARMA algorithms with all the Napoli archives. The algorithms' input had variable length between 999 to 1203 average delay values after running the average transformation. Each experiment was repeated at least 5 times to optimise parameters and to show the optimum results in terms of accuracy (NRMSE) and performance (elapsed time). Similar to phase 1, we used one third of each dataset to train and populate HDAX trends lookup table and used the remaining data for running the simulation. As shown in Table 4, HDAX compares favourably with ARMA. It shows an average accuracy of 89% while ARMA model showed 82% accurate. Moreover, HDAX shows slightly better performance in comparison to the used benchmark.

Table 4: Accuracy of HDAX and ARMA (benchmark) in the phase two of simulation runs together with speed of calculation.

MODEL	Accuracy %	Performance (sec)
HDAX	88.725941	0.014780647
ARMA	82.181929	0.016310118

5 Conclusions

This paper presented a delay forecast framework: a novel mechanism to predict delay in real-time streams by observing the delay variation and trends. The framework is part of a predictive model, which is using the forecasted QoS traces such as values of delay and jitter to predict packet loss assigned to a network node (usually an edge router). The proposed approach in this paper determines trends for variation in delay by measuring the delay variation characteristics from ongoing traffic. The motivation is to use this predicted value to indicate the current degree and severity of congestion and likelihood of packet loss, and to use it as a vital component in sender-based error and rate control mechanisms for multimedia.

We presented simulation results showing that the method can predict delay values accurately and quickly. We used the ARMA algorithm, as a benchmark, to test the accuracy and performance of HDAX in comparison to ARMA model. The HDAX algorithm showed average accuracy and speed of 89% and 0.0148 (sec), respectively.

As future work, we need to refine the algorithm to enable forecasting QoS traces with missing values. Then, the whole model in our project, including HDAX, can be installed as a software agent within a smart router on OPNET modeller to run further simulations. We will also refine HDAX metrics using statistical techniques such as estimation theory to improve the accuracy and efficiency of the algorithm. The metrics like maximum delay threshold

and window size should be determined dynamically to reflect the prevailing online status of the network. This will enable us to deploy HDAX more reliably. Finally, more formalisation is also required for specifically studying of HDAX delay time series forecaster.

References

- Batyrshin, I. Z. & Sheremetov, L. B. (2008), 'Perception-based approach to time series data mining', *Applied Soft Computing Journal* **8**(3), 1211–1221.
- Biaz, S. & Vaidya, N. H. (1998), Distinguishing congestion losses from wireless transmission losses: A negative result, in '7th Int. Conf. Computer Communications and Networks', Lafayette, LA, pp. 722–731. Proceedings of the Seventh International Conference on Computer Communications and Networks (IC3N).
- Botta, A., Dainotti, A. & Pescap, A. (2007), Multi-protocol and multi-platform traffic generation and measurement, in 'IEEE INFOCOM 2007', Anchorage, Alaska, USA.
- Box, G. E. P., Jenkins, G. M. & Reinsel, G. C. (2008), *Time series analysis: forecasting and control*, Wiley series in probability and statistics, 4th edn, John Wiley, Hoboken, N.J.
- Brakmo, L. S., O'Malley, S. W. & Peterson, L. L. (1994), 'TCP Vegas: New techniques for congestion detection and avoidance', *ACM SIGCOMM Computer Communication Review* **24**(4), 24–35.
- Bui, V., Zhu, W., Pescap, A. & Botta, A. (2007), Long horizon end-to-end delay forecasts: A multi-step-ahead hybrid approach, in '12th IEEE Symposium on Computers and Communications, 2007. ISCC 2007', IEEE, pp. 825–832.
- Carter, R. L. & Crovella, M. E. (1996), *Measuring bottleneck link speed in packet-switched networks*, Vol. 27, 28 of *Performance evaluation*, Boston University, Boston, MA, USA.
- Chatfield, C. (2001), *Time-series forecasting*, Chapman & Hall.
- Debenham, J. & Simoff, S. (2007), Informed agents: Integrating data mining and agency, in C. Boukis, L. Pnevmatikakis & L. Polymenakos, eds, 'International Federation for Information Processing—IFIP', Vol. 247, Springer-Verlag, Boston, pp. 165–173.
- Dovrolis, C., Ramanathan, P. & Moore, D. (2001), What do packet dispersion techniques measure?, in 'INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE', Vol. 2, pp. 905–914.
- Handley, M. (1997), An examination of Mbone performance, Technical report, ISI Research Report ISI/RR-97.
- Hannan, E. & Rissanen, J. (1982), 'Recursive estimation of mixed autoregressive-moving average order', *Biometrika* **69**(1), 81–94.
- Hatonen, K., Klemettinen, M., Mannila, H., Ronkainen, P. & Toivonen, H. (1996), TASA: Telecommunication alarm sequence analyzer or how to enjoy faults in your network, in 'Network Operations and Management Symposium, 1996., IEEE', Vol. 2, pp. 520–529.

- Hermanns, O. & Schuba, M. (1996), 'Performance investigations of the IP multicast architecture', *Computer Networks and ISDN Systems* **28**(4), 429–439.
- Jain, M. & Dovrolis, C. (2002), End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput, in 'Proceedings of the 2002 SIGCOMM conference 4', Vol. 32, ACM New York, NY, USA, pp. 295–308.
- Jain, R. (1989), 'A delay-based approach for congestion avoidance in interconnected heterogeneous computer networks', *SIGCOMM Comput. Commun. Rev.* **19**(5), 56–71. 74686.
- Jiang, W. & Schulzrinne, H. (2000), 'Modeling of packet loss and delay and their effects on real-time multimedia service quality'. ACM Network and Operating Systems Support for Digital Audio and Video.
- Keogh, E., Lin, J. & Fu, A. (2005), Hot SAX: Efficiently finding the most unusual time series subsequence, in 'Fifth IEEE International Conference on Data Mining (ICDM'05)', IEEE, Houston, pp. 1–8. Fifth IEEE International Conference on Data Mining.
- Klemettinen, M. (1999), A knowledge discovery methodology for telecommunication network alarm databases, PhD thesis, University of Helsinki.
- Lau, H. Y. K. & Woo, S. O. (2007), 'An agent-based dynamic routing strategy for automated material handling systems', *International Journal of Computer Integrated Manufacturing* **21**(3), 269–288.
- Miloucheva, I., Hofmann, U. & Gutierrez, P. A. A. (2003), Spatio-temporal QoS pattern analysis in large scale internet environment, in G. Ventre & R. Canonico, eds, 'Interactive Multimedia on Next Generation Networks, LNCS', Vol. 2899, Springer, Berlin, p. 282. Interactive Multimedia on Next Generation Networks: First International Workshop on Multimedia Interactive Protocols and Systems, MIPS 2003, Napoli, Italy, November 2003: Proceedings.
- Moon, S. B. (2000), Measurement and analysis of end-to-end delay and loss in the Internet, PhD thesis, University of Massachusetts, Amherst.
- Paxson, V. (1998), 'Measurements and analysis of end-to-end internet dynamics', *University of California at Berkeley, Berkeley, CA*.
- Rankin, J., Christie, G. & Kondratova, I. (2005), Mobile multimodal solutions for project closeout, in 'Proceedings of the CSCE 6th Construction Specialty Conference', Toronto, Ontario, pp. 2–4.
- Rocha-Mier, L. E., Sheremetov, L. & Batyrshin, I. (2007), 'Intelligent agents for real time data mining in telecommunications networks', *Lecture Notes in Computer Science* **4476**, 138.
- Roychoudhuri, L. & Al-Shaer, E. (2005), 'Real-time packet loss prediction based on end-to-end delay variation', *IEEE Trans. Network Service Manager* **2**(1).
- Tobe, Y., Tamura, Y., Molano, A., Ghosh, S. & Tokuda, H. (2000), Achieving moderate fairness for UDP flows by path-status classification, in 'Local Computer Networks, 2000. LCN 2000. Proceedings. 25th Annual IEEE Conference on', pp. 252–261.
- Wang, Z. & Crowcroft, J. (1991), 'A new congestion control scheme: slow start and search (Tri-S)', *SIGCOMM Comput. Commun. Rev.* **21**(1), 32–43. 116033.
- Weiss, G., Eddy, J., Weiss, S. & Dube, R. (1998), Intelligent telecommunication technologies, in L. C. Jain, R. Johnson, Y. Takefuji & L. Zadeh, eds, 'Knowledge-Based Intelligent Techniques in Industry', CRC Press, Boca Raton, pp. 249–275.
- Zadeh, L. A. (2001), 'From computing with numbers to computing with words from manipulation of measurements to manipulation of perceptions', *Annals of the New York Academy of Sciences* **929**(1), 221–252.

A Query Based Approach for Mining Evolving Graphs

Andrey Kan¹ Jeffrey Chan^{1,2} James Bailey¹ Christopher Leckie¹

¹ NICTA Victoria Research Laboratory
Department of Computer Science and Software Engineering
University of Melbourne, Australia
Email: {akan, jkcchan, jbailey, caleckie}@csse.unimelb.edu.au

² Digital Enterprise Research Institute
National University of Ireland, Galway
Ireland
Email: jkc.chan@deri.org

Abstract

An evolving graph is a graph that can change over time. Such graphs can be applied in modelling a wide range of real-world phenomena, like computer networks, social networks and protein interaction networks. This paper addresses the novel problem of querying evolving graphs using spatio-temporal patterns. In particular, we focus on answering selection queries, which can discover evolving subgraphs that satisfy both a temporal and a spatial predicate. We investigate the efficient implementation of such queries and experimentally evaluate our techniques using real-world evolving graph datasets — Internet connectivity logs and the Enron email corpus. We show that it is possible to use queries to discover meaningful events hidden in this data and demonstrate that our implementation is scalable for very large evolving graphs.

Keywords: spatio-temporal data mining, evolving graphs, dynamic graph analysis, spatio-temporal analysis, spatio-temporal query, querying evolving graphs, event discovery

1 Introduction

An evolving graph represents a graph that changes over time, which can be characterised by a series of temporal snapshots. Figure 1 shows an example evolving graph over three snapshots (time points).

Evolving graphs are very useful for modelling real world phenomena, such as communication networks, social networks, protein interactions and businesses collaboration. Some specific examples are:

- In the analysis of computer networks, the changing topology of a network can be naturally modelled as an evolving graph. A vertex may represent a workstation and an edge may indicate the existence of a communication path at a given time point. The following questions may then arise: “is there any spatially adjacent group of connections that fail synchronously?”, “which groups of connection paths are not persistent over the specified period?”. Answering such questions can facilitate fault localisation and the identification of unstable network regions.

- A corpus recording the email activity over a period of time for an organization can be analysed as an evolving graph. Vertices may represent employees and an edge indicates that an email was sent between employees during a certain time period. Questions that may arise might be “are there groups of people having their communication intensified over some window of time?”, “are there people having collaboration that lasted for approximately a month?”.
- An evolving graph approach may be used for criminal investigations (Krebs 2002, Xu and Chen 2004). Vertices represent individuals and an edge corresponds to suspected relations between two individuals. Such relationships may be inferred from different sources: phone calls, bank transfers or people being seen together. In this scenario, a criminal investigator may wish to explore the following query “what relations appear or disappear synchronously between certain groups of people and when does it happen?”.

These examples all share a common feature: there is interest in discovering matches for a specific spatio-temporal pattern. This motivates us to investigate a query based approach to mining evolving graphs.

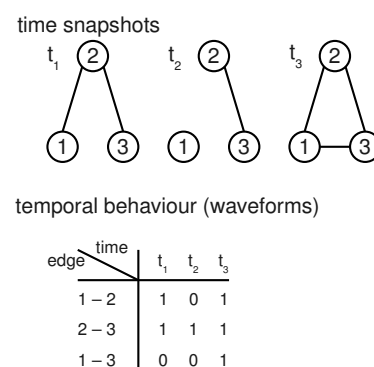


Figure 1: Sample evolving graph consisting of three snapshots. The set of vertices remains unchanged, whereas the set of edges evolves over time. Temporal behaviour of each edge can be represented by a string (waveform), where each symbol encodes the presence (“1”) or absence (“0”) of the edge at a particular time point.

To the best of our knowledge, we are the first to investigate the querying of evolving graphs by spatio-temporal patterns.

In contrast to standard database querying, querying for evolving graphs is more exploratory in nature. When undertaking analysis of an evolving graph, the user may have only a general idea of what patterns to search for. So a typical search strategy is likely to begin with the user posing an initial set of queries. The user will then iteratively analyse the obtained results and refine the queries. Since evolving graphs may be very large, it is very important that such queries can be evaluated efficiently.

There are several existing works on mining evolving graphs. Work by Chan et al. (2008) defines regions of correlated spatio-temporal changes within an evolving graph and presents an algorithm for finding all such regions. Work in Jin et al. (2007) considers a graph in which vertex labels are changing over time and presents an approach to finding frequent trend motifs, i.e., frequent subgraphs sharing the same trends in label changes. Work in Borgwardt et al. (2006) presents a method for mining frequent dynamic subgraphs, which are subgraphs that are connected, share temporal behaviour and occur at least a predefined number of times. Work in Lahiri and Berger-Wolf (2008) studies the detection of periodic behaviour in subgraphs.

These works each enumerate all occurrences (patterns) of some specified kind: inter-correlated regions, trends, frequent subgraphs or periodic behaviour. In contrast, a query based approach allows the user to specify the desired result in a specific and flexible way. For example, one might be interested only in inter-correlated regions with a particular temporal behaviour or only in frequent subgraphs that disappear at a certain time point. Since it is more specific, another advantage of a query based approach is that it is significantly faster compared to enumeration style methods.

In summary, the main contributions of our work are as follows:

- We propose a model for querying evolving graphs by spatio-temporal patterns (Section 2).
- We present two algorithms for implementing select style queries (Sections 3.1 and 3.2). We analyse the correctness, completeness, worst-case complexity and relative advantages of each algorithm.
- We run experiments on two real-world datasets (Sections 4.1 and 4.2). Experiments show that using our query model, it is possible to find subgraphs that relate to real-world events.
- We evaluate our implementations on synthetic evolving graphs (Section 4.3). Evaluation shows that querying can be implemented in an efficient and scalable manner. We found that our query implementation requires less than half a second for a graph with 10,000 changing edges.

2 Problem Statement

In this section, we formally define the problem in terms of the evolving graphs (input), the correlated subgraphs (output), and the query model that specifies the type of correlated subgraphs in which we are interested. We begin in Section 2.1 by defining the concept of an evolving graph. In Section 2.2, we define the concept of a correlated evolving graph with particular focus on how its temporal and spatial behaviour can be specified in a query. Finally, in Section 2.3 we define our query formulation and the corresponding query satisfaction problem.

2.1 Evolving Graphs

One of the main inputs for a query is an evolving graph. We begin by defining an evolving graph as well as the related concept of an evolving subgraph, based on the formulations presented by Borgwardt et al. (2006).

Definition 1. (Evolving Graph) *An evolving graph is a sequence of consecutive graph snapshots $G_{ts} \dots G_{te}$ that have the same set of vertices V , but possibly different sets of edges E_t , where $t = ts, \dots, te$. Let $E = \bigcup_{t=ts}^{te} E_t$ be a union of edges of all graphs in the sequence. We denote an evolving graph as $eg = (V, E, ts, te, \mathcal{E})$.*

\mathcal{E} is a set of strings specifying the temporal behaviour of edges. For each edge e in E , there is a string $\mathcal{E}(e)$ with symbols numbered from ts to te . If an edge e is included in snapshot G_t , then the corresponding symbol $\mathcal{E}(e)[t] = "1"$, otherwise (edge e is deleted from G_t) $\mathcal{E}(e)[t] = "0"$.

In the scope of the present work we consider only undirected, unweighted graphs and discrete time points, i.e., a sequence of graph snapshots. We focus on edge changes, assuming the set of vertices remains the same. Note that a changing set of vertices $V_t, t = ts, \dots, te$, can be replaced by a union set $V = \bigcup_{t=ts}^{te} V_t$.

Definition 2. (Evolving Subgraph) *Given two evolving graphs $eg_1 = (V_1, E_1, ts_1, te_1, \mathcal{E}_1)$ and $eg_2 = (V_2, E_2, ts_2, te_2, \mathcal{E}_2)$, eg_1 is an evolving subgraph of eg_2 if $V_1 \subseteq V_2$, $E_1 \subseteq E_2$, $[ts_1, te_1] \subseteq [ts_2, te_2]$ and \mathcal{E}_1 contains substrings from \mathcal{E}_2 , taken in the interval $[ts_1, te_1]$ for all e in E_1 . Saying that eg_1 is an evolving subgraph of eg_2 is equivalent to saying that eg_1 is included in eg_2 . This inclusion relation is denoted as $eg_1 \subseteq eg_2$.*

In the substring definition above, a substring must match a consecutive subsequence of characters, i.e., no gaps are allowed. For example, given the string "abcde", "abc" and "cd" are its substrings, whereas "ace" and "cba" are not substrings.

An evolving subgraph can be considered as a spatial subgraph of the evolving graph, extracted over some sub-period of time (see Figure 2). Note that an evolving subgraph is an evolving graph itself.

2.2 Correlated Evolving Graphs

We now consider the specific type of evolving subgraph that we seek in our query satisfaction problem, i.e., correlated evolving graphs that satisfy certain temporal and spatial characteristics. Our aim is to find evolving subgraphs whose edges exhibit a common *temporal* behaviour (in terms of insertion and deletion), as well as sharing specific *spatial* properties, such as their topological proximity.

Work by Chan et al. (2008) defined this task as an unsupervised learning problem, where the aim is to find all correlated subgraphs in a given evolving graph. In contrast, our aim is to formulate the search for correlated subgraphs as a query answering problem, where we search for correlated subgraphs that match a given query, which specifies the temporal evolution of interest.

The temporal behaviour of a changing edge can be represented by the following strings (as proposed by Chan et al. 2008). A **waveform** (denoted as W) is a string consisting of "1" and "0" symbols, reflecting the insertion or deletion of an edge from the snapshots of a graph. Symbols in the waveform are numbered starting from 1.

Another string, called the **transition sequence**, represents the overall shape of the corresponding

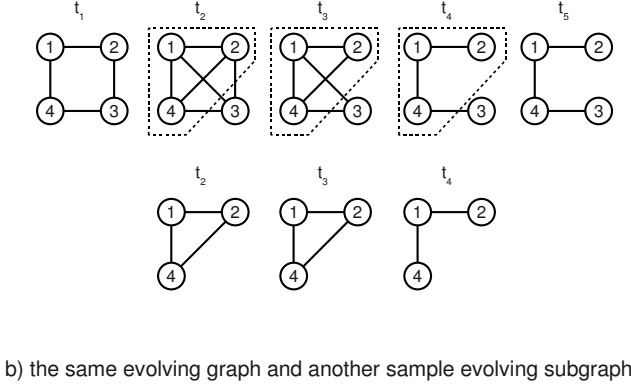
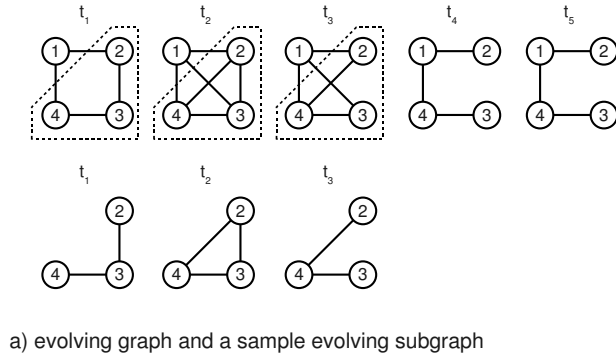


Figure 2: The figure illustrates the concept of evolving subgraph. Two sample subgraphs extracted from the same evolving graph are shown. The dotted line highlights the particular subgraph extracted.

waveform. Each occurrence of the “10” pattern in the waveform (i.e., deletion) is represented by the “-” symbol in the transition sequence, and an occurrence of the “01” pattern (i.e., addition) is represented by the “+” symbol. All other parts of the waveform are ignored. See Figure 3 for an example. We denote the transition sequence for a waveform W as $tran(W)$.

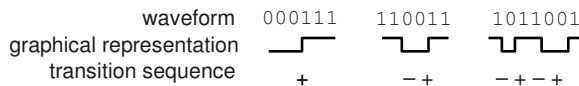


Figure 3: Sample waveforms and their transition sequences. The symbols of a waveform encode the presence (“1”) or absence (“0”) of an edge in an evolving graph at a particular time point. The symbols of a transition sequence encode a changing state of the edge: appearance (“+”) or disappearance (“-”).

In our model, the result of a query is a set of subgraphs in which all edges follow the waveform provided in the query. However, the edges in the resulting subgraphs do not need to have exactly the same waveform as in the query. It is sufficient to have similar waveforms with respect to some distance metric. We call this metric **temporal distance**. Chan et al. (2008) introduce a distance measure called “modified Euclidean distance” and argued that this measure fits well in the context of their problem statement. Since we use the same definition of waveform, we adopt the “modified Euclidean distance” in our work. Given two waveforms W_1, W_2 ,

where $length(W_1) = length(W_2) = L$

$$d_{m_euc}(W_1, W_2) = \begin{cases} 1, & \text{if } trans(W_1) \neq trans(W_2); \\ \frac{1}{L} \sum_{k=1}^L W_1[k] \oplus W_2[k], & \text{otherwise.} \end{cases}$$

Here, the operator $x_1 \oplus x_2$ represents an exclusive OR: it equals 0 if the symbols x_1, x_2 are the same, and is 1 otherwise.

This distance lies in the range $[0, 1]$, with smaller values corresponding to a better match between the waveforms. If the transition sequences differ, the highest value (complete mismatch) is returned. Otherwise the number of mismatched symbols is taken into account.

Note that strings specifying the temporal behaviour of edges in an evolving graph are essentially waveforms. The difference is that the symbols in $\mathcal{E}(e)$ are numbered starting from ts . For any edge we can obtain its waveform W_e by re-numbering the symbols in the string $\mathcal{E}(e)$.

Definition 3. (Correlated Edge) Consider an edge e of an evolving graph and the waveform W_e of the edge. Given an arbitrary waveform W and a user-defined threshold θ , the edge is correlated with W if the lengths of W and W_e are the same and the waveforms are sufficiently similar: $d(W, W_e) \leq \theta$.

Now we extend the concept of correlation to an evolving graph.

Definition 4. (Correlated Evolving Graph) An evolving graph $eg = (V, E, ts, te, \mathcal{E})$ is correlated with a waveform W if all of its edges are correlated with W and the graph $G(V, E)$ is connected. We denote the correlation to a waveform as $e \sim W$ for edges and $eg \sim W$ for evolving graphs.

Note the spatial constraint in the definition above: we require connectedness. This requirement reflects the localization property of real-world events. Another way to impose a spatial constraint is a predicate. In the context of our query model, we define the predicate as follows.

A **predicate** on an evolving graph $P(eg) = \{true|false\}$ is a function that takes an evolving graph as input and produces a Boolean value (true or false) as output. Our only requirement for a valid predicate is that it must be implementable in linear time complexity with respect to the number of vertices and edges in the input graph. We require this constraint to guarantee the efficiency of our querying algorithm (complexity analysis is presented in Section 3.3).

Predicates can be used to impose both spatial and temporal constraints. Consider an evolving graph $eg = (V, E, ts, te, \mathcal{E})$. We denote the number of vertices and edges in the graph as n_v and n_e respectively. The following are examples of possible predicates.

“Timing predicate” returns true if the evolving graph satisfy certain timing constraints:

$$P_{time}(eg) = (ts > 3 \text{ and } (te - ts) > 5).$$

“Size predicate” returns true if the evolving graph has more than N edges:

$$P_{size}(eg) = (n_e \geq N).$$

“Clique predicate” (assuming a simple undirected graph) returns true if a graph $G(V, E)$ is a clique (each vertex is connected to all other vertices):

$$P_{clique}(eg) = (n_e == n_v * (n_v - 1) / 2).$$

Note that a finite Boolean formula over the predicates is a predicate itself. For example in $P = ((P_1 \text{ and } P_2) \text{ or } P_3)$, all of P_1, P_2, P_3 and P are predicates.

We are now ready to define our main operator for performing queries.

2.3 Query Model

In our query model, the data is represented by evolving graphs, the required spatio-temporal properties are specified by waveforms and predicates, and the queries are implemented in the form of a selection operator.

Our model should be considered as a stepping stone towards a general evolving graph query language with a more sophisticated algebra. By analogy with the relational algebra, the model might be extended with operators like join and aggregation.

At present, selection is the only operator in our model. This operator is essential and our experiments show that it is sufficiently powerful to produce practically useful results.

Definition 5. (Query) A (spatio-temporal) query Q is a pair $\{W, P\}$, where W is a waveform and P is a predicate.

Definition 6. (Selection Operator) A selection operator takes as input an evolving graph eg , and a query $Q = \{W, P\}$. The output is a set of evolving subgraphs correlated with W , which satisfy P :

$$\sigma(eg, Q) = \{sg \subseteq eg : sg \sim W, P(sg) = \text{true}\}$$

We require the evolving subgraphs in the output to be maximal, i.e., there is no evolving subgraph that is included in another evolving subgraph from the same output set.

Figure 4 illustrates sample queries and their results. Recall that for a correlated evolving subgraph we require the corresponding graph $G(V, E)$ to be connected. Here V is the set of vertices, which is the same for all snapshots of an evolving subgraph, $E = \cup E_t$ is the union of edges in all snapshots.

The problem that we address in this paper is how to design an algorithm that can answer queries in a scalable manner on large evolving graphs. In the next section, we present our algorithms for implementing the queries.

3 Algorithms for Query Satisfaction

In this section, we present an efficient algorithm to implement our query selection operator. We call the algorithm “Select Basic” and describe it, along with a correctness and completeness analysis, in Section 3.1. We then present a modified version of the algorithm, which runs faster, but requires extra memory and prior indexing of an evolving graph. We call the modified version “Select Indexed” and describe it in Section 3.2. Finally we analyse the worst-case time complexity for both algorithms in Section 3.3.

A naive approach to implement a query $\sigma(eg, W, P)$ is to find all evolving subgraphs correlated with W , filter out those subgraphs that do not satisfy P , and finally select the maximal evolving subgraphs. The drawback of this approach is the computational cost of checking the inclusion relation for all pairs of evolving subgraphs in the intermediate set of results. Instead we propose the following algorithm.

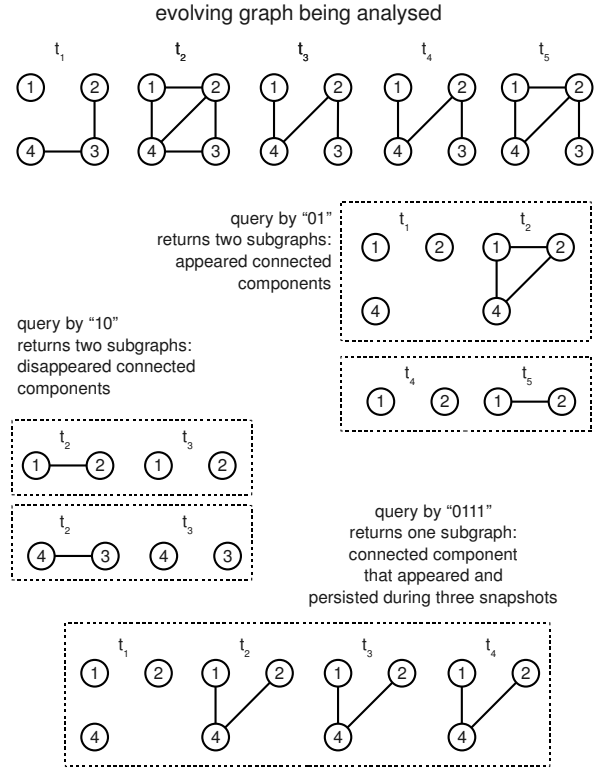


Figure 4: Results produced by selection operator

3.1 Select Basic

To implement a query we propose an algorithm called “Select Basic”. Inputs to this algorithm are an evolving graph eg , a waveform pattern W (with the length l_W) and a predicate P . The output is the result of selection $\sigma(eg, W, P)$.

The key principle of this algorithm is to select evolving subgraphs at each time point $t \in [t_s, t_e]$ independently of all other time points. The algorithm iterates over each time point and finds $sg_t \subseteq eg$, an evolving subgraph, defined on $[t, t + l_W - 1]$. sg_t includes all edges that are correlated with W over $[t, t + l_W - 1]$, and sg_t includes only such edges. After being found, sg_t is then partitioned into evolving subgraphs, such that each subgraph is correlated with W (and connected). Finally, any partitions of sg_t that do not satisfy P are filtered out.

In the “Select Basic” algorithm, a function *SubgraphWithCorrEdges()* is implemented in a direct way: it starts with empty sg_t , iterates over all edges e in E and calculates the correlation with waveform W over $[t, t + l_W - 1]$. If the correlation is sufficiently strong, e is added to sg_t .

Lemma 7. Algorithm “Select Basic” is a correct and complete implementation of the selection operator (see proof in Appendix A.1).

Considering the exploratory nature of querying, a common scenario consists of running a large number of queries over the same evolving graph. We found that it is possible to optimize this common scenario. We present a modified version of the query implementation in the following section.

3.2 Select Indexed

While algorithm “Select Basic” is computationally efficient (as shown in the next section), there are

Algorithm 1 Select Basic

Input: evolving graph $eg = (V, E, ts, te, \mathcal{E})$; waveform W ; predicate P
Output: set of matching evolving subgraphs $R = \sigma(eg, W, P)$

```

1:  $R = \emptyset$ 
2: foreach  $t$  in  $[ts, te]$ 
3:   // get subgraph containing all edges
4:   // correlated to  $W$  during period  $[t, t + l_W - 1]$ 
5:    $sg_t \leftarrow \text{SubgraphWithCorrEdges}(eg, t, W)$ 
6:   //  $P_t$  is a set of connected subgraphs
7:    $P_t \leftarrow \text{Partition}(sg_t)$ 
8:   foreach  $sg$  in  $P_t$ 
9:     if ( $P(sg)$ )
10:      add  $sg$  to  $R$ 
11:   endfor
12: endfor

```

specific application scenarios where greater efficiency can be achieved. We propose a modification of “Select Basic”, which is more efficient when multiple queries are made from the same evolving graph. The modified algorithm is called “Select Indexed”, because it uses indexing, as we discuss in detail in this section.

We found that the implementation of function *SubgraphWithCorrEdges*() is a main contributor to running time of a query. Thus we focused on optimizing this function and algorithm “Select Indexed” is a modification of “Select Basic”, using an alternative implementation of *SubgraphWithCorrEdges*() .

There are two ideas behind the optimisation. The first idea is to prune away edges that do not change over the entire evolving graph. We need to store a list of such edges, but we do not need to store and analyse their waveforms.

The second idea is to index all possible evolving subgraphs in eg in advance, and then use this precomputed data when satisfying queries. Such an indexing procedure requires additional execution time, but indexing is needed only once. Furthermore, it can be performed as a background process without requiring users to wait.

While indexing can be applied to make querying faster, the storage for indexing results requires additional memory. The amount of memory overhead is estimated in the next section.

Indexing results are stored in two hash tables called “Inner” and “Outer”. The “Inner” table is a hash table, with a waveform W as a key and the set of edges, having exactly this waveform over some period $[t, t + l_W - 1]$, as a value. The “Outer” table has a tuple $\{t, l_W; \text{tran}(W)\}$ as a key and a pointer to an “Inner” hash table as a value. A key $\{t, l_W; \text{tran}(W)\}$ points to the “Inner” table in which all keys are waveforms starting from t with length l_W and transition sequences $\text{tran}(W)$.

Consider algorithms “Prune” and “Indexing”. At first, all edges that do not change across the entire evolving graph eg are saved in a separate list and pruned from eg . Subsequent indexing is performed only on changing edges.

3.3 Complexity Analysis

The inputs to the “Select” algorithms are an evolving graph eg , a waveform W and a predicate P . The

Algorithm 2

Function *SubgraphWithCorrEdges*() in “Select Indexed”

Input: evolving graph without constant edges eg' ; list of constant edges E_C ; current time point t ; waveform W ; hash tables *Inner*, *Outer*
Output: $sg_t \subseteq eg$ is an evolving subgraph, containing all edges correlated with W over $[t, t + l_W - 1]$, and containing only such edges

```

1: // set an empty evolving graph
2: // which is defined over  $[t, t + l_W - 1]$ 
3:  $sg_t \leftarrow (\emptyset, \emptyset, t, t + l_W - 1, \mathcal{E} = \emptyset)$ 
4: if (all symbols in  $W$  equal to 1)
5:   add  $E_C$  to edges set of  $sg_t$ 
6: endif
7:  $trSeq \leftarrow$  transition sequence of  $W$ 
8:  $keyOuter \leftarrow \{t, l_W, trSeq\}$ 
9:  $Inner \leftarrow \text{Find}(Outer, keyOuter)$ 
10: if (Inner table found)
11:   foreach  $\{key; value\}$  in Inner
12:      $W_e \leftarrow key$  // a waveform
13:      $eSet \leftarrow value$  // a set of edges
14:     if ( $W$  and  $W_e$  are correlated enough)
15:       add  $eSet$  to the edges set of  $sg_t$ 
16:     endif
17:   endfor
18: endif

```

evolving graph consists of $T = te - ts + 1$ snapshots. We focus on changing edges and ignore vertices that have no adjacent edges, thus we can put $n = |V| = |E|$ (here, $E = \bigcup_{t=ts}^{te} E_t$ is a union of edges in all snapshots). A waveform has length l_W . A predicate runs in linear time with respect to the number of vertices and edges in the input graph, as we required in Section 2.2.

We now consider the worst-case complexity of the “Select Basic” algorithm. The algorithm consists of T iterations. At each iteration we apply the function $sg_t = \text{SubgraphWithCorrEdges}()$, partitioning and filtering. sg is an evolving subgraph of eg , thus the subgraph sg can have at most n vertices and n edges. Partitioning is essentially splitting into connected components and can be performed in $O(n + n)$ steps. Since we required that the predicate can be implemented in linear time (Section 2.2), each call of predicate $P(sg)$ runs in $O(n_{sg} + n_{sg})$, where n_{sg} is the number of edges in sg . After partitioning the sum of vertices and edges in all partitions is at most $(n + n)$. Therefore all calls of $P(sg)$ in one iteration run in $O(n + n)$ and overall complexity for “Select Basic” can be written as $O(T \times (n + O(\text{SubgraphWithCorrEdges})))$.

In “Select Basic” *SubgraphWithCorrEdges*() iterates over all edges in E and calculate the correlation with waveform W . Recall that we use a linear metric “modified Euclidean distance” for correlation. Thus the complexity of this function is $O(n \times l_W)$.

The total worst-case time complexity for algorithm “Select Basic” is $O(T \times n \times l_W)$, i.e., a query can be implemented in linear time with respect to its inputs, namely the parameters of the evolving graph being queried and the length of the required waveform.

“Select Indexed” differs from “Select Basic” only in the implementation of *SubgraphWithCorrEdges*() .

Algorithm 3 Prune (Constant Edges)**Input:** evolving graph $eg = (V, E, ts, te, \mathcal{E}t)$ **Output:** $eg' = eg$ without constant edges; set of constant edges E_C

```

1:  $eg' = eg$ 
2:  $E_C = \emptyset$ 
3: foreach  $e$  in  $E$ 
4:   if  $(\mathcal{E}(e)[t] = "1"$  for each  $t$  in  $[ts, te]$ )
5:     // edge is "constant"
6:     add  $e$  to  $E_C$ 
7:     remove  $e$  from  $eg'$ 
8:   endif
9: endfor

```

Algorithm 4 Indexing**Input:** $eg' = eg$ without constant edges**Output:** outer and inner hash tables $Outer, Inner$

```

1:  $Outer = \emptyset$ 
2: foreach  $e$  in  $E'$ 
3:   foreach  $l$  in  $[1, T]$ 
4:     foreach  $t_1$  in  $[ts, te]$ 
5:        $t_2 \leftarrow t_1 + l - 1$ 
6:        $trSeq \leftarrow$  tran. sequence of  $e$  over  $[t_1, t_2]$ 
7:        $keyOuter \leftarrow \{t, l, trSeq\}$ 
8:        $Inner \leftarrow FindOrCreate(Outer, keyOuter)$ 
9:        $W_e \leftarrow$  waveform of  $e$  over  $[t_1, t_2]$ 
10:       $keyInner \leftarrow \{W_e\}$ 
11:      // set of edges that have
12:      // the same waveform over  $[t_1, t_2]$ 
13:       $eSet \leftarrow FindOrCreate(Inner, keyInner)$ 
14:      add  $e$  to  $eSet$ 
15:    endifor
16:  endifor
17: endfor

```

In the optimised implementation of this function in "Select Indexed" the time complexity is determined by the number of insertions of edges into sg_t (lines 5 and 15) and the number of correlation computations (line 14). In the worst case, the evolving subgraph sg_t can have as many edges as eg and correlations may need to be calculated for every single edge. In this case, function *SubgraphWithCorrEdges()* has time complexity $O(n \times l_W)$ and algorithm "Select Indexed" has the same time complexity $O(T \times n \times l_W)$ as the "Select Basic".

However, in practice the number of edges in sg_t tends to be much less than n , because we expect different groups of edges to follow different behaviours at some time point. Furthermore, the computation of correlation occurs only once per group of edges that have the same transition sequence. Therefore we expected that the complexity of the optimised *SubgraphWithCorrEdges()* function would be on average $O(k \times l_W)$, where $k \ll n$ is some constant. Under this assumption, the complexity of "Select Indexed" is $O(T \times k \times l_W)$.

"Select Indexed" requires prior preprocessing ("Prune" and "Indexing") of an evolving graph. The worst-case time complexity of the algorithm "Prune"

is $O(T \times n)$.

Now consider one iteration of the inner loop in the algorithm "Indexing". Computing a transition sequence can be performed incrementally as time progresses. Searching a hash table is considered as a pseudo-constant operation. Therefore the overall worst-case time complexity of algorithm "Indexing" is $O(T \times T \times n)$.

Storing the indexing data (hash tables) requires additional space. For each waveform W that occurred at a certain time point and has a certain length, there can be no more than n' matching edges, where n' is the number of edges that experienced at least one change in the evolving graph eg . The total number of these unique waveforms is $(T \times T)$, i.e., a waveform can occur at any time point in $t \in [ts, te]$ and have a length $l \in [1, T]$. Therefore the overall memory complexity is $O(T \times T \times n')$.

In the next section, we compare the relative advantages of each of our two algorithms in terms of their execution time on various types of datasets.

4 Experimental Evaluation

In this section we evaluate the effectiveness and efficiency of our proposed algorithms on a range of real-world and synthetic datasets. The aims of our experiments have been: i) to evaluate the practical feasibility of querying evolving graphs and gauge the interestingness of the results, and ii) to measure the computational scalability of our algorithms for querying large graphs.

To evaluate the practical feasibility of querying, we have analysed evolving graphs built from two real-world datasets (Sections 4.1 and 4.2). The first dataset consists of snapshots of the routing topology of the backbone of the Internet. This dataset was collected by Chan et al. (2008). The second dataset is the Enron email corpus as presented by Klimt and Yang (2004). These datasets are particularly interesting because there are known real-world events that we expect to be reflected in the data. Furthermore, there have been other studies that have analyzed the same datasets from different perspectives. Thus we can compare and contrast our findings with these related works.

To measure the computational scalability, we have evaluated our algorithms on synthetic evolving graphs (Section 4.3). We generated random evolving graphs with different sizes and measured the running times required for querying each graph.

We also compare our results with findings on the same datasets from related works (Section 4.4).

4.1 Evaluation on Internet BGP Routing Topology Graph

Our first experiment was the analysis of a part of the Internet backbone routing topology. At the backbone level, the Internet comprises a set of Autonomous Systems (ASs), each of which is a network under a single controlling authority. The Border Gateway Protocol (BGP) is responsible for establishing routes between these ASs. From a high level perspective, an AS can be considered as a node in the Internet connectivity graph, where each AS has a unique number. Therefore the Internet can be represented as an evolving graph, where a vertex corresponds to an AS and is labeled by its AS number. An edge in such a graph corresponds to an existing routing path (connectivity) between two ASs. The topology of the Internet may change over time, thus the graph is evolving.

Chan et al. (2008) have constructed an evolving graph from the AS connectivity logs. In their work they present the details of building the graph and argue for the difficulty and importance of analysing changing edges of the graph. We have used a copy of their graph in our experiment.

The evolving graph represents only the US part of the Internet, i.e., each vertex corresponds to an AS registered in the USA. The graph consists of around 10,000 vertices and 18,000 edges. Only about 700 edges are changing, while others persist in the graph during the whole time period. The graph spans over the 41 time snapshots, which we number starting from one. Snapshots are taken in two hour intervals. The first snapshot is taken on the 28 August 2005 at 1 pm and the last on the 31 August 2005 at 10 pm (times are in UTC format).

In August 2005 there were Hurricane Katrina landfalls in some southern US states. Hurricane Katrina was a major event, which is known to severely affect the Internet infrastructure in the region of the landfalls (Cowie et al. 2005). The second landfall of Katrina occurred in Louisiana approximately between snapshots 12 and 13 (August 29, between 10 and 11 am UTC).

We were interested in the major effects of Katrina landfalls, thus we filtered out small results using the predicate of our query. For all queries we used a predicate that returns true for evolving graphs containing more than 3 edges.

Two hour snapshots provide quite a fine temporal granularity. Therefore the consequences of the same event may appear in neighbouring snapshots. To address this, for all queries we set the correlation threshold as 0.2. This means, we allowed 80% to 100% correlation between found subgraphs and a waveform in a query, rather than requiring exact match.

We used three different temporal patterns. The first pattern is “111000”. It was designed to discover failure regions, more specifically, connections that were stable for several snapshots and then disappear for an extended period. We did not use the “10” pattern, because this query might result in additional subgraphs corresponding to random short term network failures. (Of course one can use “10” or any other pattern, depending on the graph being analysed and the particular purpose of the analysis. For example, we later use the “01” pattern in our experiment on the Enron dataset.)

The second pattern is “000111”. Subgraphs with such temporal behaviour can be considered as recovery regions.

The last pattern is “110011” and it corresponds to a failure, followed by a prompt recovery. For each query we obtained a set of subgraphs and manually reviewed the largest subgraphs in each set.

Recall that each vertex represents an AS with a unique numerical identifier. Chan et al. (2008) show that by using the AS number it is possible to retrieve two types of information: the US state where the AS is registered (and most likely physically located), and the organization to whom the AS belongs. We used this information when presenting our results in Figure 5. For selected vertices either the US state or organizational affiliation is shown. Waveforms in the figure show the temporal behaviour of the majority of the edges in the corresponding subgraph.

Evolving graph A most likely corresponds to a network failure due to the second landfall of Hurricane Katrina: all edges of the graph are connected to ASs in Louisiana and the timing of failure matches the timing of the landfall. Cowie et al. (2005) reported that a large percentage of destroyed networks around 2 pm UTC (snapshot 13) were in Louisiana.

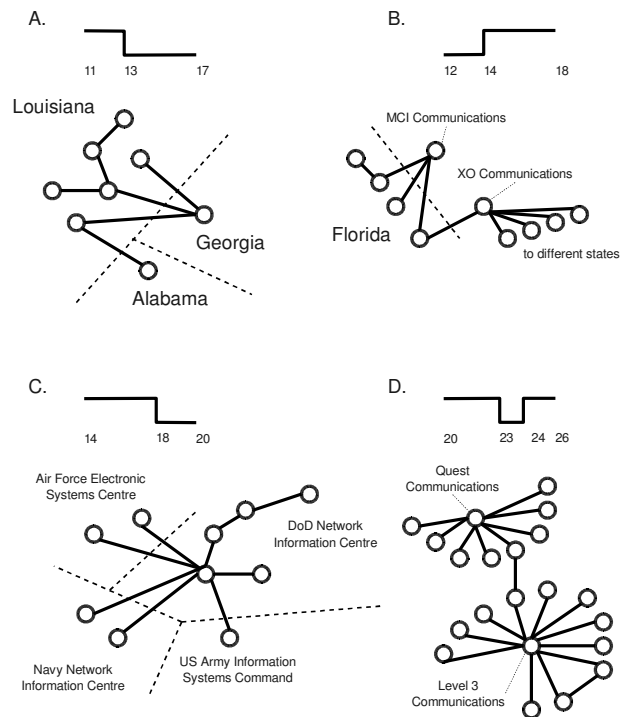


Figure 5: Some of the results from querying the AS connectivity graph. The results are interpretable, i.e., each found subgraph can be related to a real world event. For selected vertices either the US state or organizational affiliation is shown. Waveforms represent the behaviour of the majority of the edges in a subgraph. Time stamps are snapshot numbers.

Evolving graph B represents a recovery region. There was an earlier landfall of Katrina in Florida. Although the timing of this landfall is outside of the period we analyse, we note the connections with Florida and suggest that graph B represents the recovery of major communication networks from this earlier strike.

Other interesting events are represented by evolving graphs C and D. Evolving graph C represents a failure of a segment of a military network: all its ASs are registered within the Department of Defence, Navy and Air Forces. Graph D corresponds to a failure and a quick recovery of two major Internet Service Providers.

We can clearly associate four distinct events to each of the graphs A — D. Note that graphs A and B were not merged together, despite being spatially adjacent and graphs A and C were not merged, despite having similar temporal behaviour. This demonstrates the importance of both spatial and temporal components of queries in order to separate the events properly.

In this section, we have described experiments on the AS connectivity evolving graph. Our work shows that using several queries, we are able to find subgraphs that can be related to known real-world events. In the next section we describe an experiment on another real-world dataset.

4.2 Evaluation on Enron Email Corpus

The Enron email corpus is a large set of emails collected from the Enron corporation over 3.5 years (Klimt and Yang 2004). The dataset was used in a number of studies and has been analysed from different perspectives (Diesner and Carley 2005,

Berry and Browne 2005, Rowe et al. 2007, Borgwardt et al. 2006). We built an evolving graph from the Enron corpus and used querying to analyse this dataset.

Each email in the dataset contains sender and recipients addresses and a time stamp. Borgwardt et al. (2006) turned the Enron dataset into an evolving graph. They divided the total period in which emails were collected into a number of intervals. They represented employees as vertices and put an edge in a particular snapshot if there was an email between the corresponding employees in this time interval.

The graph of Borgwardt et al. contains only 15 snapshots, which results in a large time interval per snapshot. We were not satisfied with this granularity and we built an evolving graph that is different in several ways.

First, snapshots in our graphs are taken in one week intervals.

Second, we decided to restrict the analysis to the year 2001. Shetty and Adibi (2004) reported that the number of emails is not equally distributed over time, and there is a much larger number of emails in 2001 compared to other years. In addition, it is known that this year was particularly rich in events for the Enron organization. Examples of the events are the California power crisis and the bankruptcy of Enron.

Third, we used a threshold for the number of emails. We expected an event in an organization to be reflected by a “higher than usual” email traffic between employees related to the event. Therefore we calculated the average number of emails sent within one week between two employees and set a threshold above this number. In the i^{th} snapshot of our evolving graph we put an edge if there were more than 3 emails sent between the corresponding employees within the i^{th} week.

Since there are duplicate emails in the dataset, we first pruned any duplicates. The evolving graph we built consists of 140 vertices and 244 edges. All edges are changing, i.e., there is no edge that exists in all snapshots. The graph has in total 52 snapshots (52 weeks of the year 2001).

In order to demonstrate our flexibility in constructing queries, we used three temporal patterns that were not used for the AS connectivity graph. We filtered out small results with a predicate returning true, if a graph contains more than 3 edges. We expected email discussions of the same event to have mostly synchronous starting times, i.e., within the same week. Thus we required results to be 100% synchronised within a pattern, by setting the correlation threshold to zero.

The result of each query is a set of evolving subgraphs. We manually reviewed the largest subgraphs. Each evolving subgraph may be related to event(s) in the organization, and vertices represent employees related to the event(s). Each edge corresponds to a number of emails. We analysed the contents of these emails in order to understand what event(s) they might correspond to. We present the evolving subgraphs found by different temporal patterns in Table 1. The first pattern we used is “01”. It can be interpreted as an “event occurrence”. One of the evolving subgraphs found corresponds to emails sent within week 4 (late January 2001). There was an electricity crisis in California in 2000 — 2001. In mid January 2001 a blackout affected hundreds of thousands of citizens. This blackout was followed by a declaration of a state of emergency by Governor Davis. Enron was one of the largest energy companies in the USA at that time. Therefore it is not surprising to find this event in the emails communication of

Enron employees. Subjects or bodies of almost all emails of the evolving subgraph are related to the political and other consequences of the electricity crisis in California.

Another waveform we used for querying is “0110”. This kind of behaviour can be interpreted as short term cooperation: average email traffic, followed by intensification for two weeks and decreasing afterwards. One of the graphs found indicates that such a pattern occurred in mid August 2001. A review of the emails corresponding to the subgraph showed that there were two main discussion topics. The first topic relates to a conference call with Portland. This can be concluded from some email subjects and the following text found within the emails: “... conference call with Portland”, “I have organized ... groups to attend Portland for one week ...”. The second relates to some activities related to “Cash/Prompt”. Several emails have “Cash/Prompt” keyword and one of the emails says: “I have decided to move Frank Ermis from his role as Prompt/Season ...”.

Lastly, in a “mid term cooperation” query (“011110”), one of the found evolving subgraphs is related to the daily “TRV” reports which were distributed during several weeks.

Note the variety of temporal behaviours for which we can search. We can encode different situations of interest (e.g., “short term cooperation”) with different waveforms. Note also the comprehensiveness of the discovered information: for each subgraph we can infer the main discussion topics (event), the time of pattern occurrence and the list of participants in the discussion.

In summary, our experiment on the Enron corpus shows that, as in the case with the AS connectivity graph, we are able to find evolving subgraphs that can be related to known real-world events. The following section presents our evaluations of running time for querying synthetic graphs.

4.3 Evaluation on Synthetic Graphs

The purpose of our evaluation on synthetic graphs has been to analyse how the query execution time varies under different conditions.

There are a number of factors that can affect the execution time for a query. In Section 3 we introduced two algorithms for implementing a query. We report the results of experiments on synthetic graphs for both algorithms, where appropriate.


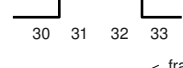

Another factor affecting the execution time is the size of the evolving graph being analyzed. The size can be expressed as the number of vertices, edges and time snapshots. In Section 3.3 we showed that the number of snapshots and the number of edges contribute to the time complexity in a similar manner, and the number of vertices is bounded by the number of edges. Therefore in synthetic graphs, we vary only the number of edges.

The last factor that we considered is the length of a waveform in a query. The worst-case time complexity for the algorithms is $O(T \times n \times l_S)$, thus we expected a linear increase in execution time for longer waveforms. In our evaluation we used several waveforms with lengths varying from 2 to 32.

We generated random evolving graphs with the number of edges varying from 50 to 100,000. All edges are changing. Each graph has $T = 100$ snapshots. After generating the graphs we modified some edges in order to guarantee that for the queries that we try, the result is non-empty.

We implemented both algorithms in C++ and ran our queries on a PC with Intel(R) Core(TM)2 Duo E8400 @ 3GHz CPU and 3GB RAM. We repeated

Table 1: Some of the results from querying the evolving email graph. The results can be related to real world events.

Query pattern and its interpretation	Found evolving subgraphs (week numbers shown)	Selected subjects of emails, corresponding to the edges	Related event(s)
01 — "event occurrence"	late January  james.steffes kay.mann jeffrey.hodge david.delaney elizabeth.sager john.lavorato keith.holst mike.grigsby phillip.allen richard.sanders jeff.dasovich	Cheney: White House Mtg Mon On Calif. Pwr Prob...; Edison's Filing in District Court in L.A.; Governor Davis; Governor Davis names advisors; New Bill Introduced in CA Legislature.	California power crisis
0110 — "short term cooperation"	mid August  mike.grigsby frank.ermis keith.holst jason.wolfe matt.smith	West Power ...; FW: West Power Rotation; FW: Western Strategy Briefing; Frank Ermis and Matt Lenhart; Cash/Prompt and Prompt/ Season Traders.	"West Power" conference call with Portland; activities within "Prompt/Season"
011110 — "mid term cooperation"	late September — October  errol.mclaughlin larry.may dutch.quigley mike.maggi john.arnold	TRV Notification: (NG - PROPT P/L - 09/28/2001); TRV Notification: (NG - PROPT P/L - 10/01/2001); TRV Notification: (NG - PROPT P/L - 10/03/2001); TRV Notification: (NG - PROPT P/L - 10/04/2001); TRV Notification: (NG - PROPT P/L - 10/11/2001).	daily "TRV" reports

each execution time measurement several times and report averaged values.

Recall that the "Select Indexed" requires prior indexing of an evolving graph. Therefore, at first, we analysed the execution times and memory consumption required for indexing evolving graphs of different sizes. Results are presented in Table 2.

The results show that there is a roughly linear dependency of both execution time and memory consumption on the number of edges. Execution time values remain within reasonable limits. Furthermore the indexing can be performed as a background process and thus does not require an operator to wait.

In contrast, memory consumption is a critical issue. We found that when an evolving graph has more than 5,000 changing edges, indexing information requires more than 1GB of storage. We did not apply the "Select Indexed" for larger graphs, because it required more memory than the maximum file size allowed in our operating system.

Note that in the "Indexing" algorithm, only the number of changing edges contribute to the running time and memory consumption, because constant edges are pruned away before indexing. In our synthetic evaluation all edges are changing, i.e., we found that the memory consumption is an issue for graphs with more than 5,000 *changing edges*. This is quite a large number. For reference, the AS connectivity graph that we analysed contains 700 changing edges. The evolving graph we built from the Enron corpus has 244 changing edges. Furthermore, for larger graphs we still can use the "Select Basic", which does not require indexing.

The next experiment aimed to analyse the dependency of query execution time on the size of the evolving graph. Results are presented in Figure 6. We measured execution times for both algorithms on the synthetic graphs with 50 — 5,000 edges (Figure 6.a)

Table 2: Execution times and memory consumption for indexing evolving graphs with different sizes (algorithm "Indexing"). For each graph $T = 100$

Number of edges	Time (sec)	Memory (MB)
50	4	11
100	7	22
500	37	111
1,000	70	220
5,000	360	1,125

and for the "Select Basic" only on graphs with 10,000 — 100,000 edges (Figure 6.b). In all queries we used the same waveform "01010101".

The results show that "Select Indexed", when applicable, significantly outperforms "Select Basic". Another conclusion that can be drawn is that both algorithms demonstrate good scalability: roughly linear dependency of the running time on the number of edges (note the quasi-logarithmic scale of x axes).

In the last experiment we evaluated the running times for different query waveforms on the same evolving graph. We used a synthetic graph with $|E| = 1,000$ edges and $T = 100$ snapshots. Waveform lengths vary from 2 to 32. The results are presented in Figure 7.

According to the worst-case time complexity analysis in Section 3.3, running times should be longer for longer waveform lengths. In contrast, experimental results show that measured running times decrease for both algorithms. We explain this phenomenon as follows. In practice the running time is less than it would be in the worst case. This time depends on the number of the results found by the query. On average, for longer patterns there are fewer results. Therefore for longer waveforms we observe

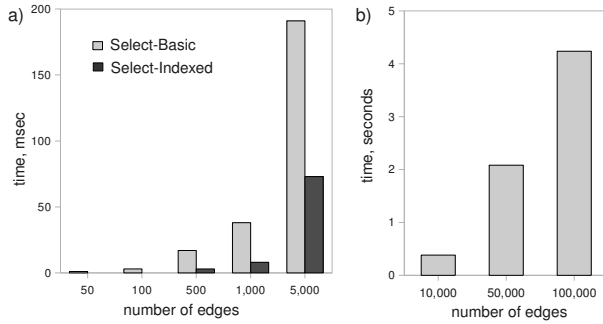


Figure 6: Execution times for querying graphs with different numbers of edges and the same number of snapshots $T = 100$. In all cases the same waveform “01010101” was used for querying. In a) two algorithms are compared. In b) only “Select Basic” is used. Note the quasi-logarithmic scale of the x axes.

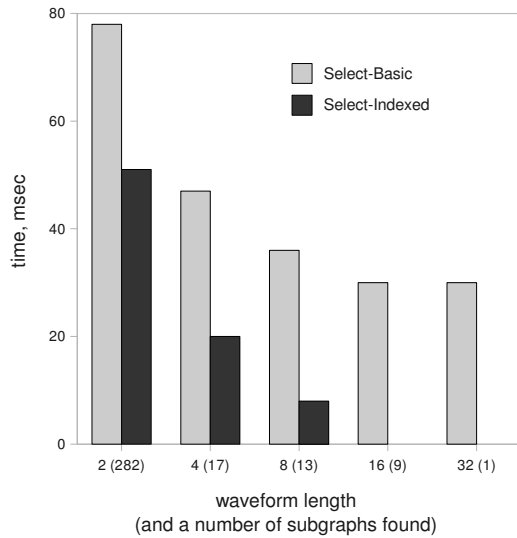


Figure 7: Execution times for different query waveforms. Two algorithms are tested. Queries are performed from the same evolving graphs with $|E| = 1,000$ edges and $T = 100$ snapshots. Note the logarithmic scale of the x axis.

decreased running times.

In summary, our evaluation on synthetic graphs demonstrates several properties of our algorithms. First, “Select Indexed” can significantly outperform “Select Basic”. On graphs with more than 5,000 changing edges, memory consumption is a critical issue for indexing. For such graphs “Select Basic” can still be used.

Second, both algorithms demonstrate good scalability: the execution time increases roughly linearly with the size of the graph.

Last, in contrast with the theoretical worst-case analysis, longer waveforms in queries result in faster query execution. This happens because for longer patterns there are fewer matching results and, in practice, running time depends more on the number of results found by a query, then on the waveform length.

The following section presents comparing and contrasting our findings on two real-world datasets with the results of previous studies.

4.4 Results Discussion

Some of the related studies report evaluation results on the same real-world datasets as we used in our experiments. This allows us to compare and contrast our findings with the work of others.

Our first experiment was performed using the Internet AS connectivity evolving graph. Chan et al. (2008) report their findings on this evolving graph. Comparing our findings, we note that by using three queries we are able to find subgraphs corresponding to the all the regions that they reported.

Consider also the performance evaluation on synthetic datasets in the work of Chan et al. (2008) and in our work. A query can be executed ten thousand times faster than enumerating all inter-correlated regions from the same graph. This is because finding inter-correlated regions essentially results in a large set of all possible query results. If we are interested in regions with a particular temporal behaviour, we need to search this set itself. In contrast, we can directly search for a required waveform by using querying.

The second dataset that we used is the Enron email corpus. There have been many works that analysed this dataset. Borgwardt et al. (2006) constructed an evolving graph from the dataset. They searched for frequent dynamic subgraphs. However they report only quantitative results, without relating the found subgraphs to events or persons.

The results of Diesner and Carley (2005) provide a deeper insight into the organizational structure of the Enron corporation. In particular they report the list of “key players” in the organization. In our results “key players” can be inferred from the subgraph topology (see Table 1). For example, from our results, we note that James Steffes and Mike Grigsby can be considered as key persons. Diesner et al. also included these employees in their list of “key players”. Note that in our study we are also able to elaborate in what particular situations these people played an important role. Diesner et al. do not present such an analysis. They also do not provide references to concrete events or discussion topics.

Berry and Browne (2005) analysed the contents of emails using non-negative matrix factorization. They report several discovered topics of discussion. Some of the topics match our findings (e.g., the California power crisis), while others differ. This can happen due to the following reasons. First, the Enron corpus consists of emails systematically collected for a subset of employees. A number of emails were sent to or received from people outside of this subset. Our purpose has been a demonstration of the capabilities of querying, rather than thorough investigation of the Enron case. Thus, in our analysis, we omitted emails that have either the sender or the recipient outside of the set of employees, for which the dataset was collected.

Second, we were searching for topics developing according to a temporal pattern. In contrast, the method of Berry et al. returns all topics, from which they reviewed the largest. For example, one of the topics we have found is the “TRV” reports. Berry et al. do not report this topic, apparently because it was not large enough. However “TRV” reports might be of particular interest as an example of mid-term cooperation patterns in Enron.

Regarding the computational efficiency, note that Berry et al. presented a method in which the input length is determined by the number of symbols in all emails. In contrast, the input length for querying is bounded by the number of emails.

In summary, our experiments on real-world and synthetic datasets demonstrate that:

- querying evolving graphs can discover real-world events, reflected in the datasets;
- querying can be performed within a reasonable amount of time even on large graphs (hundreds of milliseconds for a graph with 10,000 edges);
- performance degrades roughly linearly as the size of the evolving graph grows;
- a single query runs roughly ten thousand times faster than an approach that enumerates all inter-correlated regions (which was the approach used by Chan et al. 2008);
- querying is capable of identifying discussion topics, without the need for analysis of the contents of all emails in the Enron corpus.

5 Related Work

We categorize related work by the field of study. Our work focuses on mining evolving graphs by querying them with spatio-temporal patterns. Therefore, the main fields related to our study are: “mining evolving graphs”, “spatio-temporal patterns” and “querying”.

Mining evolving graphs by different spatio-temporal patterns has been addressed by a number of works. The pattern proposed by Chan et al. (2008) is a region of correlated spatio-temporal changes. This region is an evolving subgraph in which all edges experience similar temporal behaviour. Lahiri and Berger-Wolf (2008) “propose a new mining problem of finding periodic or near periodic subgraphs in dynamic social networks”. Jin et al. (2007) focus on evolving graphs with changing weights. They propose a pattern called “trend motif occurrence”. This is essentially a connected subgraph, in which all vertices have decreasing or increasing weights. Borgwardt et al. (2006) aim to search for frequent dynamic subgraphs, i.e., multiple occurrences of identical topological subgraphs with the same temporal behaviour.

Spatio-temporal patterns are used not only for mining evolving graphs, but also for other datasets. For example, Celik et al. (2006) introduce a new co-occurrence pattern as frequent spatio-temporal co-location of objects with different types. They propose a method for finding such patterns in a general spatio-temporal dataset. This method does not address graphs specifically. Hadjieleftheriou et al. (2005) propose a more flexible pattern: an ordered list of spatial predicates. The order (exact or relative) in the list is a temporal predicate. They describe a framework for querying such patterns from a set of trajectories. They do not consider particular issues of querying evolving graphs.

An evolving graph can be thought of as a database of strings of “0” and “1”. Querying substrings over a string database is a much studied field. Many in-memory and disk based algorithms were proposed (e.g., Kahveci and Singh, 2001, Meek et al., 2003). Usually a string edit distance and its variations are used to measure similarity between strings. However in the context of spatio-temporal queries, similarity is measured by temporal distance, using waveforms and transition sequences of compared strings. Thus it is not a trivial task to apply algorithms based on the string edit distance to querying evolving graphs. Furthermore, the spatial dimension (graph topology) is not considered in these algorithms.

Querying static graphs has been studied by a number of researchers. Zhang et al. (2009) focus on efficient searching of required topological pattern occurrences in very large graphs. He and Singh

(2008) propose a formal language for querying and manipulating graphs. They presented a graph algebra as an extension of relational algebra. They also address some challenges of pattern matching. Trissl and Leser (2006) introduce an index structure to facilitate reachability and distance queries in a graph.

The studies that address querying graphs disregard the fact that graphs may change over time. In other words, these studies do not consider evolving graphs.

The works related to mining evolving graphs do directly address the challenges and opportunities provided by temporal change. These works aim to search for different spatio-temporal patterns: inter-correlated region, periodic behaviour, trend motif occurrence, and frequent dynamic subgraphs. Each of these patterns has potentially useful practical applications. However, given the large variety of application domains and research questions one might have when mining evolving graphs, a more flexible way of defining patterns is of great interest. Therefore we introduced a framework that allows a user to define a spatio-temporal constraint according to the needs of a particular analysis.

6 Conclusions and Future Work

We have presented a novel approach for mining evolving graphs: queries for user-defined spatio-temporal patterns. Users can specify the required temporal and spatial constraints and search for matching evolving subgraphs. We formally posed the problem of querying by spatio-temporal patterns. This problem was addressed by two algorithms that implement the query: the first one is a straightforward implementation and the second uses indexing. Using indexing allows us to execute queries faster, but this optimization is achieved with the cost of increased memory consumption.

We evaluated our approach on two real-world datasets: the Internet AS connectivity graph and the Enron email corpus. We also ran experiments on synthetic evolving graphs. We discussed the obtained results and related them to other studies that used the same datasets.

We summarize the **importance of our findings** as follows:

- we evaluated querying on the datasets that relate to very common real life phenomena: Internet topology and emails in an organization;
- the datasets studied are large (up to 700 changing edges), which makes manual analysis of changes infeasible; thus there is a need for automated analysis tools;
- we are able to discover real-world events, reflected in the data, using querying;
- our implementation is fast and scalable: a query runs for hundreds of milliseconds for a graph with 10,000 edges and the running time increases roughly linearly as the size of a graph grows.

These promising results inspire us to continue our work on querying along the following directions for **future work**:

- consider operators for manipulating query results and extend our query model to a richer query language for evolving graphs;
- consider evolving graphs with weighted and directed edges; for example, for the email corpus, it would be possible to use directed edges and weight them according to the email traffic volume.

Acknowledgement

This work is partially supported by National ICT Australia and Science Foundation Ireland (SFI) under CLIQUE Strategic Cluster, grant number 08/SRC/I1407. National ICT Australia is founded by the Australian Government's Backing Australia's Ability initiative, in part through the Australian Research Council.

References

- Berry, M. and Browne, M. (2005), 'Email surveillance using non-negative matrix factorization', *Computational & Mathematical Organization Theory* **11**(3), 249–264.
- Borgwardt, K., Kriegel, H., Wackersreuther, P. and Munich, G. (2006), Pattern mining in frequent dynamic subgraphs, in 'Proceedings of the 6th International Conference on Data Mining', pp. 818–822.
- Celik, M., Shekhar, S., Rogers, J., Shine, J. and Yoo, J. (2006), Mixed-Drove Spatio-Temporal Co-occurrence Pattern Mining: A Summary of Results, in 'Proceedings of the 6th International Conference on Data Mining', pp. 119–128.
- Chan, J., Bailey, J. and Leckie, C. (2008), 'Discovering correlated spatio-temporal changes in evolving graphs', *Knowledge and Information Systems* **16**(1), 53–96.
- Cowie, J., Popescu, A. and Underwood, T. (2005), Impact of Hurricane Katrina on Internet Infrastructure, Technical report, Renesys Corporation.
- Diesner, J. and Carley, K. (2005), Exploration of communication networks from the enron email corpus, in 'Proceedings of Workshop on Link Analysis, Counterterrorism and Security, SIAM International Conference on Data Mining', pp. 21–23.
- Hadjieleftheriou, M., Kollios, G., Bakalov, P. and Tsotras, V. (2005), Complex spatio-temporal pattern queries, in 'Proceedings of the 31st International Conference on Very Large Data Bases', VLDB Endowment, pp. 877–888.
- He, H. and Singh, A. (2008), Graphs-at-a-time: query language and access methods for graph databases, in 'Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data', ACM New York, NY, USA, pp. 405–418.
- Jin, R., McCallen, S. and Almaas, E. (2007), Trend motif: A graph mining approach for analysis of dynamic complex networks, in 'Proceedings of the 7th IEEE International Conference on Data Mining', IEEE Computer Society Washington, DC, USA, pp. 541–546.
- Kahveci, T. and Singh, A. (2001), An efficient index structure for string databases, in 'Proceedings of the International Conference on Very Large Data Bases', Citeseer, pp. 351–360.
- Klimt, B. and Yang, Y. (2004), 'The enron corpus: A new dataset for email classification research', *Lecture Notes in Computer Science* **3201**, 217–226.
- Krebs, V. (2002), 'Mapping networks of terrorist cells', *Connections* **24**(3), 43–52.
- Lahiri, M. and Berger-Wolf, T. (2008), Mining Periodic Behavior in Dynamic Social Networks, in 'Proceedings of the 8th IEEE International Conference on Data Mining', IEEE Computer Society Washington, DC, USA, pp. 373–382.
- Meek, C., Patel, J. and Kasetty, S. (2003), Oasis: An online and accurate technique for local-alignment searches on biological sequences, in 'Proceedings of the 29th international conference on Very large data bases-Volume 29', VLDB Endowment, pp. 910–921.
- Rowe, R., Creamer, G., Hershkop, S. and Stolfo, S. (2007), Automated social hierarchy detection through email network analysis, in 'Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web Mining and Social Network Analysis', ACM New York, NY, USA, pp. 109–117.
- Shetty, J. and Adibi, J. (2004), The Enron email dataset database schema and brief statistical report, Technical report, University of Southern California.
- Trissl, S. and Leser, U. (2006), GRIPP Indexing and Querying Graphs Based on Pre- and Postorder Numbering, Technical Report 207, Humboldt-Universitaet zu Berlin.
- Xu, J. and Chen, H. (2004), 'Fighting organized crimes: using shortest-path algorithms to identify associations in criminal networks', *Decision Support Systems* **38**(3), 473–487.
- Zhang, S., Li, S. and Yang, J. (2009), GADDI: distance index based subgraph matching in biological networks, in 'Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology', ACM New York, NY, USA, pp. 192–203.

Appendix A

A.1 Proof of Lemma: Correctness and Completeness of Algorithm "Select Basic"

Consider two time points $i, j : i \neq j$ and two evolving subgraphs sg_i and sg_j , containing all edges correlated with W over $[i, i + l_W - 1]$ and $[j, j + l_W - 1]$ respectively, and containing only such edges. None of the evolving subgraphs of sg_i is included in any of the evolving subgraphs of sg_j , because their temporal intervals are not included in one another. Furthermore, partitioning (which is essentially splitting into connected components) produces a set of non-overlapping and connected evolving graphs. Therefore, the evolving graphs in the output are maximal. The partitioning does not affect the temporal characteristics, thus the result of partitioning is a set of evolving subgraphs that are correlated with W . After the filtering step, the evolving subgraphs in the results set satisfy P . Therefore all evolving subgraphs in R satisfy the selection constraints and algorithm "Select Basic" is correct.

We refer to an evolving subgraph $sg \subseteq eg$, that is correlated with W over $[t, t + l_W - 1]$ and satisfies P as an *eligible subgraph*. Consider an eligible subgraph sg , which is not included in any other eligible subgraph. In algorithm "Select Basic" sg is included in sg_t (line 5). Then, after partitioning sg_t (line 7), the evolving subgraph sg becomes one of the partitions and this partition is not filtered out (line 9). Therefore all maximal eligible evolving subgraphs are included in the set of results and algorithm "Select Basic" is complete. \square

Mining Minimal Constrained Flow Cycles from Complex Transaction Data

Meng Xu¹

Michael Bain²

¹ School of Computer Science and Engineering
University of New South Wales,
Email: mxux967@cse.unsw.edu.au

² School of Computer Science and Engineering
University of New South Wales,
Email: mike@cse.unsw.edu.au

Abstract

Transaction data in domains such as trading records from online financial or other markets, logistics delivery registers, and many others, are being accumulated at an increasing rate. In this type of data each transaction has a complex format, being usually associated with attributes such as time, numerical quantity, parties involved, and so on. Performing data mining on trace record data of complex transactions may enable the extraction of knowledge about implicit relationships which will benefit the community in different ways, for example by improving market efficiency and oversight, or detecting scheduling bottlenecks. However, the size of data sets of this type is usually enormous, and therefore in order to perform searching or mining techniques considerations of efficiency are often more important than correctness. In this paper we develop a framework to embed different methods to speed up search algorithms with the goal of detecting cycles in trace record data that fit a given constraint predicate on the amount of transaction quantities that can flow in a direction. The method is shown to improve significantly on a naïve approach, and suggests a number of directions for further work.

Keywords: transactions, data mining, search, cycles, flow, cluster.

1 Introduction

Transaction data describe interaction events carried out within a group of agents. Typical examples include information gathered from supermarkets or stock exchanges. Data are stored on an individual event basis; each event contains all the necessary fields to describe it. Most of the data sets will have a dimension of time, a numerical value such as purchased quantity or total amount, the parties involved in the transaction, which could be a single participant or a two-sided participant pair, i.e., sender/receiver. Typically each event is stored separately, but to obtain useful information a series of similar events needs to be combined.

By converting the transaction data into a directed graph, we may be able to apply graph mining techniques to extract useful information. The typical type of structure we are looking for is one or more *cycles*

that satisfy a user-supplied predicate or set of predefined constraints. Typically also we require that the solutions be found according to some *search bias* or order, so for example, in this paper we assume that the smallest or minimal cycle (fewest vertices) should be found first.

Abnormal cycles in the graph sometimes can have special meanings, for instance, indicating that someone is engaging in criminal behaviour, or a new type of fashion is appearing in the population. This kind of technique can be adopted into surveillance domains to help the system to pick up unexpected behaviors. Typical areas that this can be applied to is financial market surveillance or national security. For instance, if a large amount of money flowed out from a client and after certain time same amount of money flowed back, and the client repeated this for several times, this could mean the client is doing some form of money laundering. Another example could be where a person sent out many packages to a certain factory in a foreign country, and got the all packages back from overseas just before a terrorist attack, which may indicate that this person is a member of the responsible terrorist organisation.

Mining different types of constrained cycle from graphs has been quite well studied in the past (for example, see (Chakrabarti & Faloutsos 2006), (Yoshida et al. 1994), (Washio & Motoda 2003), (Yamada & Kinoshita 2002)), and many algorithms have been proposed. Although some of the methods are claimed to be able to find cycles efficiently, i.e., in a reasonable amount of time, the complexity of most of these algorithms still depends heavily on the structure and complexity of the graph. Typically scalability is an issue for the kind of transaction data in which we are interested, since the runtime performance tends to drop dramatically once input graphs get really dense.

In this paper, we propose a framework which has the capacity to process highly connected transaction graphs to search for the kind of constrained flow cycles of interest. We have implemented an algorithm designed to apply to transaction data from financial markets, where millions of transactions happen between fewer than hundreds of market traders, searching for cycles that show differences from normal trade patterns. The remainder of the paper is organized as follows. Section 2 discusses related work and Section 3 contains the problem definition. In Section 4 we analyze one type of transaction data of interest, and the corresponding transaction graph. Section 5 contains our initial solution to the problem of finding constrained flow cycles used on this data and Section 6 contains experimental results from our implementation. Section 7 has conclusions and suggestions for further work.

Copyright ©2009, Australian Computer Society, Inc. This paper appeared at the Eighth Australasian Data Mining Conference (AusDM 2009), Melbourne, Australia. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 101, Paul J. Kennedy, Kok-Leong Ong and Peter Christen, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

2 Related Work

The demand for mining structured data, especially graphs, has increased dramatically in the past decade. Most of this research has focused on finding subgraphs from massive graph data, where the subgraphs have special topological meaning. Several approaches have been proposed based on greedy search (Washio & Motoda 2003), notably SUBDUE (Cook & Holder 1994) and Graph Based Induction (GBI) (Yoshida et al. 1994). In these approaches infeasible computation is avoided by employing greedy search on the data set, which however results in incomplete searching results. Other methods developed later on have used different approaches, for instance, inductive logic programming as in WARMR (Dehaspe & Toivonen 1999) to obtain a form of completeness of the result set by employing monotonicity properties of subsumption as is common in data mining. In these approaches the search space can be constrained by the user, for example, by using a language bias and employing support and confidence type thresholds.

The type of subgraph we are looking for in transaction graphs is a particular cycle where constraints are not only focused on topology but also the weight and direction of the edges. Finding cycles in a graph is a well-explored area, and there are many existing algorithms that are designed for finding different types of cycles. In the 1970s Tiernan proposed an efficient algorithm EC (Tiernan 1970) which is capable of finding all elementary circuits in a graph. However the runtime still depends on the size and complexity of the graph. Yamada and Kinoshita (Yamada & Kinoshita 2002) devised a method to explore negative cycles in a directed graph, where the word “negative” denotes that the total (numerical) weight on edges in the cycle is negative.

Graph mining in stock trade transaction data is still a relatively new area to explore. As far as we are aware, there is no existing method specifically designed to detect constrained cycles in stock trade transaction data. However, Palshikar and Apte (Palshikar & Apte 2008) have applied graph clustering technique to detect what they term “collusion sets”. It uses an unsupervised learning framework to detect possible clusters in the graph where the total volume traded within the cluster is much higher than outside. The algorithm is oriented to find dense subgraphs from what they term a “stock flow graph”. However, since this is essentially a graph of aggregated transactions it cannot find trading cycles that satisfy other constraints, since they are effectively hidden inside the aggregated graph due to loss of information on individual transactions.

3 Problem Definition

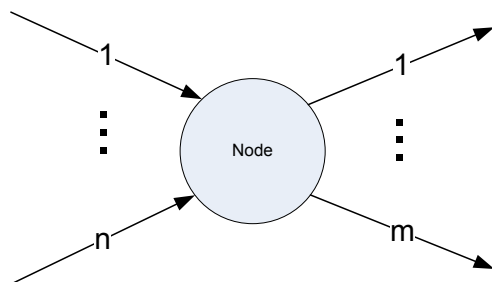


Figure 1: Partial Cycles.

The original problem is derived from markets, such as a stock exchange, where different forms of techniques can be used for the purpose of manipulating the market. Market prices can be influenced by either buying or selling significant amounts of stock. In order to drive the price up or down, a group of traders will trade a large quantity of stocks around between themselves. As the purpose of this type of behaviour is to move the price only, they need to keep their net position equal to zero. The way they accomplish this is to trade the stock around between themselves in a pattern which looks like a form of cycle.

In order to achieve the goal of manipulating the price, stock can be traded together as one enormous trade which may attract attention from surveillance departments. The way to avoid that is to split the trade into a sequence of small trades. Since surveillance systems often use a numerical threshold, such as volume, to detect abnormal behaviours, a sequence of consecutive small trades makes this very difficult for the system to detect. The number of similar legitimate trades will cause the system to generate too many false positives when the threshold cutoff is lowered to the level that the system can pick up these small consecutive trades.

Therefore the problem can be divided into two steps. The first step is to generate all possible candidates, and the second step is to try to eliminate as many false positive as possible. To develop our framework we will assume the domain of stock market trades as the basis to construct a transaction graph, although this framework is general enough to apply also to a wide class of other types of transaction data.

3.1 Search Graph Construction

Before applying a search algorithm, the data needs to be converted into a graph-based data structure. The way to achieve this is to construct a separate directed-parallel graph for each stock S , where vertices (nodes) are the traders who have participated in trading this stock, and edges are individual trade transaction events, with the direction from seller to buyer labelled to indicate the amount of stock “flowing” from one to another. Since the edges represent trades and traders can trade many times with each other, we allow parallel edges in the graph, although each label is of course unique.

Definition a directed-parallel graph G is a set $\{V, E\}$, where nodes V is the set of participants involved in the transaction data, and directed edges E represent the set of each individual transaction in the data set. The direction of the edge is a one-to-one mapping starting from the initiator node to the receiver node, $I \rightarrow R$ where I is the initiator and R is the receiver.

Since this representation contains all the relevant information of the original entire transaction record, the graph G can be extremely complex, or dense. The word complex here refers to strongly connected. For example, in one stock trading data set we selected, the number of nodes V is only 64 for a particular day, but the number of edges E exceeds 8000, which means each node V_i has more than 125 edges connected on average. The potential number of cycles we are facing could be excessive under the following assumption:

Assumption In Figure 1, each node has n in-edges and m out-edges. The maximum number of potential partial cycles including this node that can be formed by the *in/out* edge pairs is $n * m$.

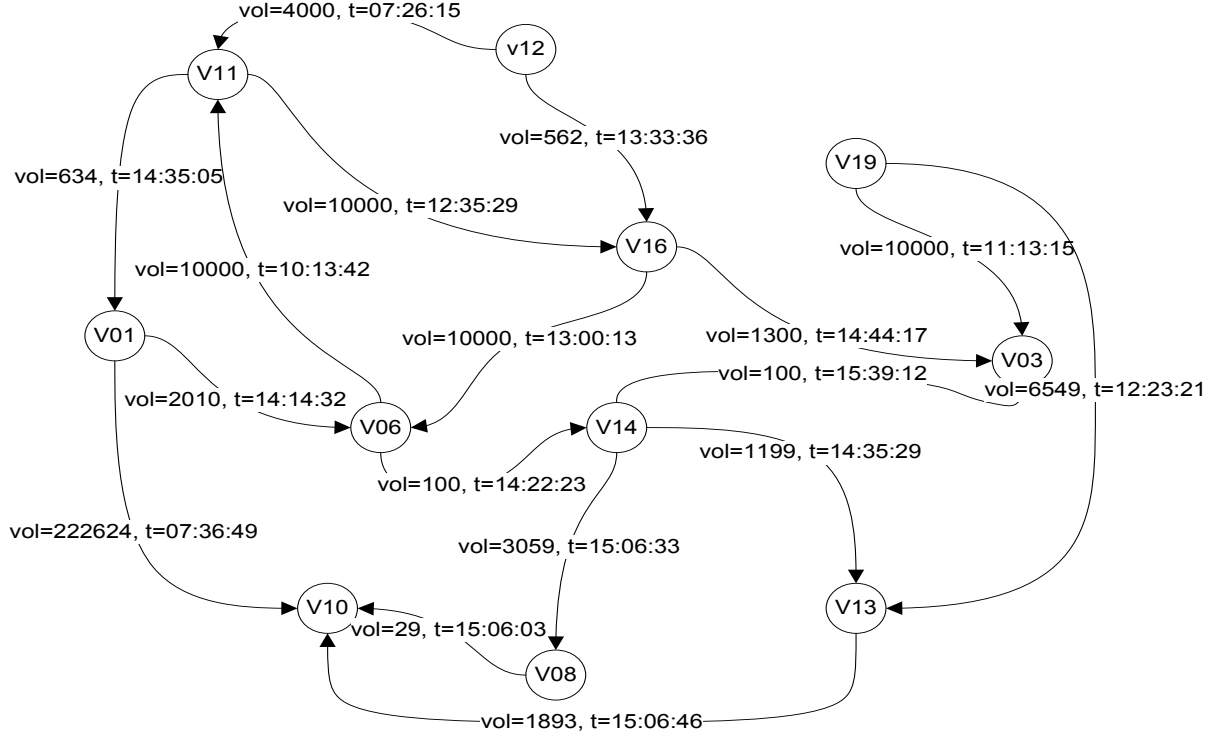


Figure 2: Transaction Graph.

3.2 Constrained Flow Cycles

The definition of a cycle in a directed graph can vary depending on different types of constraints. Some cycles may have specific constraints on the total weight of the edges, while others may have requirements on the number of nodes and edges. The type of cycle we are looking for here is one where the weights on every edge have to be approximately the same, and the order of time attributes in the corresponding label set has to align with the direction of the associated edges. Hence we adopt the term “flow” since this gives the impression that a certain weight persisting along the edge labels of the cycle looks like a volume of water flowing in the direction of the cycle.

Definition a cycle C consists of a set of edges $e_i = \{e_1 \dots e_n\} \subseteq E$. Each e_i is

$$e_i = \{t_i, w_i\}$$

where t_i is the time attribute of e_i , and w_i is the weight of the edge e_i . The cycle flow must satisfy the following constraints:

1. the i_{th} transaction e_i needs to occur before $i+1_{th}$ transaction e_{i+1} , $t_i < t_{i+1}$.
2. the weight w_i on each edge e_i has to be approximately the same, $w_i \approx w_{i+1}$.

4 Construction of the transaction graph

The data set used in this paper to analyze the problem is from a stock exchange. The reason to choose this data set is because the characteristics of the trading data allow us to construct the graph which has the properties of a small number of nodes and a large number of edges. Although many people buy and sell stocks in the stock market, exchanges do not allow individuals to trade directly, thus anyone who wants to

Stock ID	Date	Time	Price	Volume	Value	Buyer	Seller
----------	------	------	-------	--------	-------	-------	--------

Figure 3: Format of the transaction data set.

buy or sell shares needs to go through a registered broker, where the number of brokers in a particular stock exchange is fixed. This property limits the number of participants in the data set by only focusing on the broker level, in other words the number of nodes in the graph will be limited. By selecting a volatile day where a huge number of trades happened, a highly connected graph with limited number of nodes can be constructed.

We assume that the transaction data from stock trading system is in the form shown in Figure 3. The meaning of each attribute is as follows:

1. *StockID* is the identifier of the traded stock.
2. Combined *Date* and *Time* field contains the time stamp for each trade.
3. *Price* is the market price when the trade occurs.
4. *Volume* is the number of shares traded.
5. *Value* is the total amount of money this trade realized.
6. *Buyer* is the buy side trader.
7. *Seller* is the sell side trader.

However, this is just a small portion of the actual data set. The original data set will have more fields than above, but for the purposes of this research unnecessary fields are removed.

From the data set, a transaction graph will be constructed where each trade represents an edge with

a trader on each end as the nodes of the graph.

Definition a transaction graph is a directed graph constructed on a per stock basis (different stocks will have different transaction graphs) denoted as $G_{stock} = (V, E, a_e)$ where V is the set of traders labeled by trader ID. E are the directed edges with a set of attributes a_e attached. The set of attributes $a_e = (vol, t)$ has attributes vol as the number of shares traded and t the time stamp of the trade.

As the number of outgoing edges for a particular trader is normally more than 100. Shown in Figure 2 is a simplified example of a constructed transaction graph. It consists of 11 nodes and 17 edges. Each edge represents a trade, for instance, the edge from v12 to v11 illustrates that a trade happened between trader v11 and v12, where v11 bought 4000 shares at time 07:26:15 from v12. An example of a constrained flow trading cycle in this graph would be v06, v11 and v16, where 10000 shares flows along the edge direction from v06 to v11 then to v16 and back to v06.

Clearly, any approach based on exhaustively searching the transaction graph will suffer from a combinatorial explosion. Also, the notion of heavy trading still needs to be specified by the user. Below we present polynomial time algorithms to identify candidate collusion sets.

5 Naïve Search Approach

An obvious naïve search approach is just an exhaustive depth first search, where a complete set of potential candidates is generated from the original search space.

Algorithm 1 Naïve Search

Input : Transaction graph \mathcal{G} , depth limit dl , and a constraint predicate on paths in \mathcal{G}
Output : list of cycle_paths c_list

```

c_list =  $\emptyset$ 
for every node  $v$  in the  $\mathcal{G}$  do
  apply Depth-First Search from  $v$ , with maximum
  depth limit set to  $dl$ ;
  insert each new candidate path into  $c\_list$  if it
  satisfies the constraint predicate;
end for

```

Although the resulting set covers all the possible candidates, it is likely that too many false positives will be produced. The performance also suffers from massive computational requirements, and is thus inapplicable to real situations.

6 Proposed Solution

As discussed in the previous section, the graph can become extremely complex with such a small number of nodes compared to the large number of edges. It is almost impossible to search the entire search space within a reasonable time. Due to the characteristically large number of in and out edges for each node, the potential search space grows exponentially with the complexity of $\mathcal{O}(c^n)$, where c is the estimated potential partial cycle for each node.

Therefore we consider that a heuristic approach is necessary. The method we are proposing consists of two parts, a clustering-based “classifier” followed by a depth-first search algorithm. The objective of the first stage is to quickly identify a subset of potential edges for inclusion in the set of candidates. In the second stage a search can be conducted to a greater depth to enable better coverage of potential cycles.

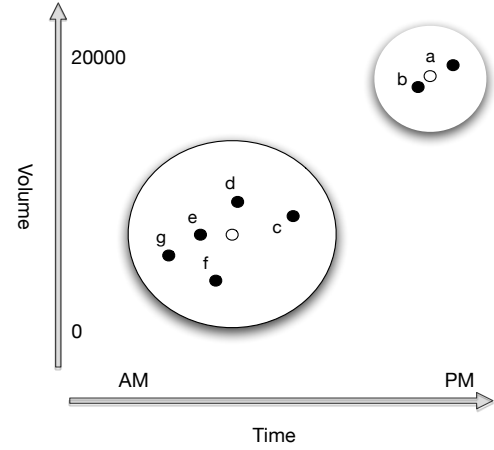


Figure 4: Clustering example.

6.1 Clustering Classifier

The classifier is based on clustering each individual transaction of the initiating agent node. Every node will have its own such classifier, the objective of which is to determine whether the current transaction occurs due to normal market behaviour or not. The classifier will be trained on each transaction involving an initiator. Since the number of agents is not large relative to the number of edges in the transaction graph this means the number of classifiers to be learned is practical.

Consider the attributes of each transaction in vector format:

$$T = \{time, price, volume, value, buyer, seller, \dots\}$$

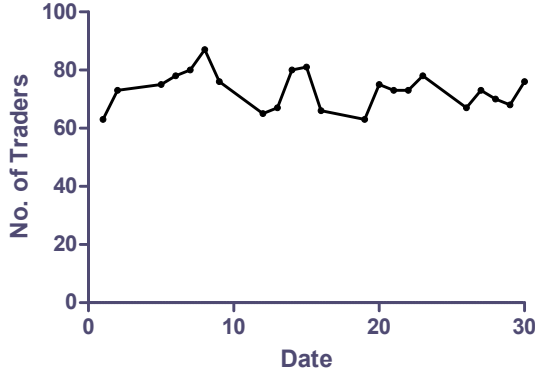
A Euclidean distance measure is adopted to measure the distance between two transactions, with weights w_i associated with each attribute difference to prioritise the key attributes while eliminating noise:

$$\Delta T = \{w_1 * \Delta time, w_2 * \Delta price, w_3 * \Delta volume, \dots\}$$

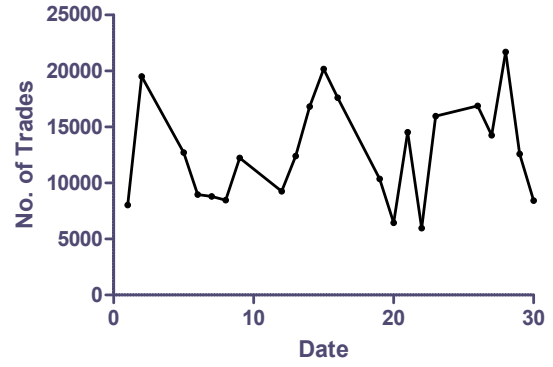
The weights w_i are obtained from domain knowledge together with performance metrics. Here the domain knowledge refers to knowledge from domain experts, for example how important is the volume attribute compared to price. In fact here price is not as relevant compared to other more important factors, such as volume. Time is also important, where assumptions can be made by experts, for example, that abnormal behavior may well happen at market opening, where massive activity helps to cover up the transactions, or during lunch time when the market is not as volatile, and hence it may be easier to control market prices.

Performance metrics here means that by adjusting the weight of a parameter the run time and number of possible results can be controlled by the user.

For the purposes of this paper, only attributes *time*, *volume* and *buyer* are used, therefore w_3 , w_4 and w_6 will be set to 0. The distance between two transactions are determined by the sum of attribute differences with w_i . For non-scaleable attributes such as *time*, the difference is measured in seconds. Meanwhile the attribute *buyer* will be kept as a separate measurement, which ranks how often the seller sells stock to each buyer.



Number of active traders for a month.



Number of trades for a month.

Figure 5: Behaviours in real trading data in a typical stock.

6.2 Clustering Algorithm

The idea behind the clustering algorithm is to try to identify the outliers of the overall trading distribution quickly. As abnormal trades happen rarely compared to normal trades, this algorithm is designed to exclude these “majority class” behaviours from the cluster without incurring any major loss of efficiency. For this reason a straightforward heuristic methods was adopted rather than using a more standard clustering algorithm.

Algorithm 2 Clustering Algorithm

Input : distance threshold, transaction graph \mathcal{G}

Output : single cluster c

```

cluster  $c = \emptyset$ ;
while  $size(c) < threshold$  do
  if  $c = \emptyset$  then
    choose a node  $v$  at random from  $\mathcal{G}$  as the centre
    of the cluster;
     $c = c \cup \{v\}$ ;
  end if

```

Set the centroid to be the new centre of the cluster, then increase the radius by unit distance;

```

  if new neighbours found within radius then
    add new neighbours to  $c$ ;
  else
     $c \leftarrow \emptyset$ 
  end if
  if tried every node then
    increase unit distance;
  end if
end while

```

output cluster c

The method will determine a single cluster containing the “typical” behaviours of the agent, resulting in a certain number of instances centered on a randomly selected initial seed instance and bounded according to a distance parameter pre-set by user. It starts by picking an instance randomly from the space as the centre of the cluster. The cluster grows a unit distance every time if new instances are included, until reaches the preset threshold. If no new instance is included, a new centre for the cluster will be chosen.

The example illustrated in Figure 4 shows the general idea of the clustering. Here black nodes are trades

performed by a particular trader. As can be seen in the example most of the trades happens in the morning and the volumes are relatively small.

To find the cluster, the following steps are performed:

- node a is chosen at random as the starting point. Since only one node is included, the centre of the cluster remains unchanged.
- the radius is increased by unit distance, therefore node b is included, updating the centre of the cluster.
- the radius is increased by unit distance again, but no new nodes are included.
- random node g is chosen as the new cluster centre.
- keep increasing the radius while new nodes are included, until one big cluster with nodes g , e , f , d and c is formed.

6.3 Search algorithm

The search algorithm is constructed based on depth first search (DFS). By specifying the depth limit $size_limit$ and threshold value $threshold$, the algorithm will find all possible flow cycles within the depth limit and threshold value. We also implemented and tested a breadth-first based alternative, but this was not found to give any advantage in this setting.

The DFS algorithm performs depth first search, starting from each initiator (trader) each node. According to the connectivity of the graph, assuming the average number of outgoing edges for a node is approximately m and the depth limit is set to n , then the potential number of searches we are facing is about m^n . Therefore the algorithm uses the above selection criteria, together with the cluster classifier which identifies abnormal behaviors, to reduce the search space. The algorithm only performs search on selected edges which are passed from the classifier.

Note that the algorithm may produce duplicate results for different initiator nodes since the search is performed on a per node basis. To check for this, any cycles found by the search algorithm for a set of traders will be inserted into the output list if and only if the output list does not already contain the same group of traders with identical edges. The output list is then ordered in increasing size of trader set.

To ensure that the cycles found are potential candidates, a predicate is used to determine at each stage

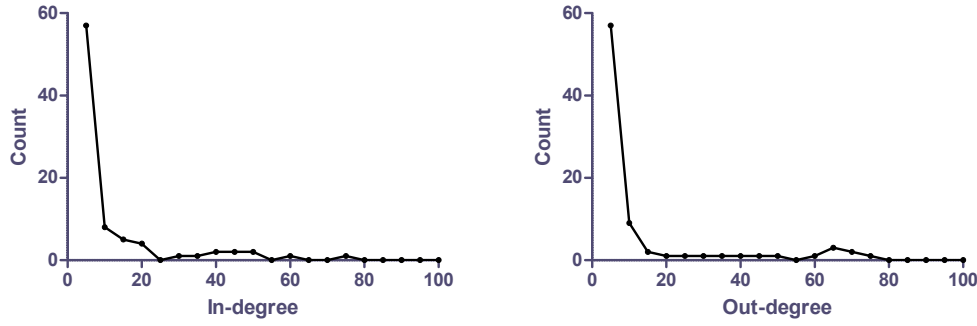


Figure 6: Graph Connectivity Distribution

Algorithm 3 Depth-first search based algorithm**Input :** sizelimit, threshold**Output :** list of cycle_paths **c_list**

```

for every node  $v$  in the graph do
  input to cluster classifier
  prevpath  $p\_path \leftarrow \emptyset$ 
  Def search(node  $v$ ) {
    for every out-edge  $e_i$  of  $v$  do
      if  $last\_time\_in\_p\_path \leq e_i.time$  then
        if  $diff(e_i.weight, avg(p\_path.weight)) \leq threshold$  then
           $p\_path \leftarrow e_i$ 
          if  $e_i.to == v$  then
            if  $p\_path$  contains abnormal edge then
               $p\_path \rightarrow c\_list$ 
               $p\_path \leftarrow \emptyset$ 
            end if
          end if
          if  $p\_path.size() \geq sizelimit$  then
            break;
          end if
          search( $e_i.from$ )
        end if
      end if
    end for
  }
end for

```

$$where \text{diff}(w_1, w_2) = \frac{abs(w_1 - w_2)}{avg(w_1, w_2)}$$

of the search expansion that new nodes fit the constraints. In Algorithm 3 this is implemented by the *diff* predicate on pairs of weights. This is designed to ensure that any cycle produced has each edge such that an approximately zero net weight “flow” will be obtained.

7 Results

Since the problem of detecting such cycles is inherently difficult we were not able to find a source of data containing already known examples. Additionally, such data is often subject to confidentiality restrictions. We were however able to obtain data for typical trading behaviour. Our experimental approach was therefore to examine the real data for general properties of traders and trading activity as shown in Figure 5, and properties of the trading graph as shown in Figure 6. Based on this we generated synthetic data to test our approach, and the results we report were based on that.

7.1 Synthetic Data Generation

Different sizes of graphs that preserve the property of real data sets are needed for testing purposes. By analyzing a real data set for a period of one month, the majority number of active traders involved in every single day falls between 60 to 80 (left of Figure 5), and the number of trades are approximately $4k \sim 20k$ (right of Figure 5).

In graph theory, besides the number of trades (edges) and traders (nodes), connectivity is one of the most important basic characteristics that determines the properties of the graph. By summarizing the number of in-degree and out-degree edges for every node for a period of time, we will get a general idea of how nodes are connected within the graph. i.e., some of the nodes may have very low degree, which may refer to long term investors, and some highly active nodes, indicating more aggressive traders.

By plotting the connectivity distribution of the graph, observation shows that the data set actually follows a power law distribution (Figure 6). In order to model this property, we have adopted the BA model proposed by Barabási and Albert (Barabási A.-L and Albert, R. 1999). The BA model consists of two parts, *Growth* and *Preferential attachment*. The algorithm keeps adding new nodes and new edges to the existing graph, until the required number of nodes is added. When a new edge is added between a new node and existing nodes, *preferential attachment* decides which existing node it should connect to, with a probability which is proportional to the in/out degree of that node. This model follows the “rich get richer” idea, which best replicates the situation in the stock data set where some traders have a fairly high in/out degree compared to other traders.

Using this model, we generated various sizes of synthetic graph by setting different numbers of node $n = \{30, 60, 100\}$ and edges $e = \{100, 500, 1000, 2000, 5000, 10000, 50000, 100000\}$. For the rest of the attributes such as time, price and volume, random values were generated according to a normal distribution. True positive test cases are manually inserted into synthetic data set. The attribute values of test cases are set outside the clusters.

7.2 Experimental Results

We have performed experiments on different sizes of graphs with different numbers of node and edge combinations, together with variation of the input parameters. We varied the volume threshold by $\{1\%, 5\%, 10\%\}$ with an upper-bound of cycle size of 10, and 12 true positive test cases were inserted into the data set.

node = 60, csz = 10 with 12 test cases inserted

Threshold =	0.01		0.05		0.1	
No. of Edges	No. of Cycles Found	Test Case Found	No. of Cycles Found	Test Case Found	No. of Cycles Found	Test Case Found
100	7	4	7	7	13	9
500	9	6	18	10	21	6
1000	11	5	17	7	35	8
2000	13	7	28	9	42	10
5000	17	8	58	11	79	11
10000	24	9	75	10	120	10
50000	48	10	120	11	308	12
100000	61	9	348	10	1033	12

Figure 7: Search Results for 60 nodes and csz = 10

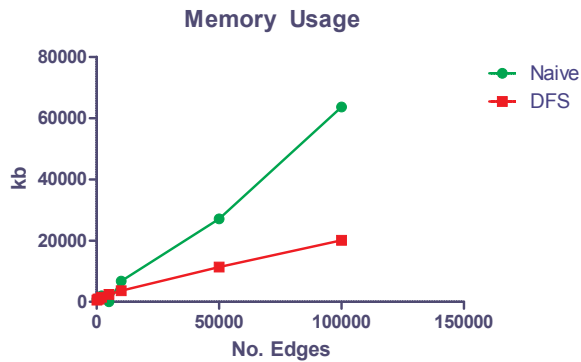


Figure 8: Memory Usage Comparison.

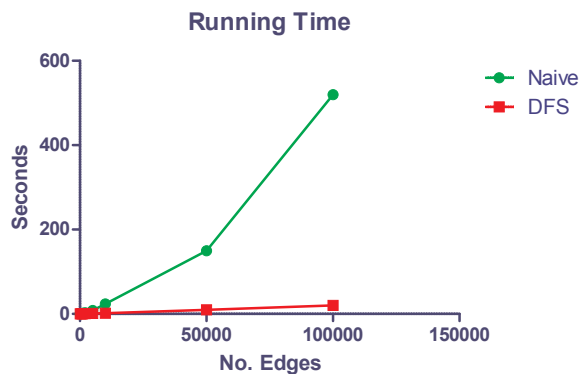


Figure 9: Run Time Comparison.

Figure 7 shows the number of possible flow cycles found by the algorithm, where graphs all have 60 nodes but different numbers of edges. We set the upper-bound of cycle size to be 10, and varied the weight threshold to get different results. Meanwhile the threshold of the cluster boundary was fixed at 80%. As shown on the above table, the number of possible flow cycles tends to grow as either the size of graph or the weight threshold increases.

Artificial flow cycles were inserted into the test data. The algorithm was able to detect some of the test cases, with additional candidates “discovered” from the data itself. As the table shows, the number of artificial test cases found tends to increase as the number of edges increases. The reason for this behaviour is that the cluster is better trained to identify irregular behaviours as the number of trades increases.

In terms of performance, the algorithm delivers much better performance than the naïve approach where no filters are used, in terms of both memory usage and run time. Figure 8 & 9 illustrate the differences when number of node, maximum cycle size and threshold are fixed. The gap between two methods tends to increase as the number of edges increases.

8 Conclusions and Further Work

In this paper we introduce the problem of mining flow cycles in a dense directed graph, subject to a given set of constraints, and with an ordering requirement on the output. On this problem a naïve approach is impractical for real world data sets. Hence we introduced customized, heuristic algorithms to find such cycles within the transaction graph as defined.

The clustering method we adopted was based on domain properties, but it is quite general in the sense of relying only on a suitable definition of atypicality or “unusualness” in the transaction trace data. As usual in clustering approaches, the choice of distance function is critical, and further work is needed on this. However initial results from our implementa-

tion demonstrated a clear improvement over a naïve approach, as expected.

Once filtering has been done by the clustering stages of the approach then a non-informed depth-first search is implemented to detect the cycles. The definition of a constraint predicate on flow through the cycle is required and is executed during the search. We adopted a measure that allowed some variation in the flow between successive edges to allow for noise in the data. Nonetheless, this measure will enable the detection of complex patterns whilst ensuring an approximately net zero flow through the cycle.

However, this paper only provides a framework of how such a search based approach can be efficiently achieved. The correctness of the algorithm still highly depends on the clustering component, and one next step will be to try and improve the to deliver more accurate results. The complexity of the problem also needs to be properly characterized. Further work is also required to specify more precisely the role of constraint predicate evaluation in this setting, and how this interacts with the clustering pre-processing step.

The DFS algorithm also need to be improved to enable the ability of discovering “split” edges, where one particular edge is divided into a number of parallel edges with sum equivalent to the original. E.g., trader 1 sold 100 shares to trader 2, and trader 2 traded with trader 1 twice with 50 shares each. Additionally, more work in the direction of exploiting known techniques of search and representation bias in data mining, for example towards minimality of mined cycles, is required.

References

- Barabási A.-L and Albert, R. (1999), ‘Emergence of Scaling in Random Networks’, *Science* **286**(5439), 509–512.
- Chakrabarti, D. & Faloutsos, C. (2006), ‘Graph mining: Laws, generators, and algorithms’, *ACM Comput. Surv.* **38**(1), 2.
- Cook, D. J. & Holder, L. B. (1994), ‘Substructure discovery using minimum description length and background knowledge’, *CoRR* **cs.AI/9402102**.
- Dehaspe, L. & Toivonen, H. (1999), ‘Discovery of frequent datalog patterns’, *Data Mining and Knowledge Discovery* **3**, 7–36.
- Palshikar, G. K. & Apte, M. M. (2008), ‘Collusion set detection using graph clustering’, *Data Mining and Knowledge Discovery* **Volume 16**(2), 135–164.
- Tiernan, J. C. (1970), ‘An efficient search algorithm to find the elementary circuits of a graph’, *Commun. ACM* **13**(12), 722–726.
- Washio, T. & Motoda, H. (2003), ‘State of the art of graph-based data mining’, *SIGKDD Explor. Newsl.* **5**(1), 59–68.
- Yamada, T. & Kinoshita, H. (2002), ‘Finding all the negative cycles in a directed graph’, *Discrete Applied Mathematics* **118**(3), 279–291.
- Yoshida, K., Motoda, H. & Indurkha, N. (1994), ‘Graph-based induction as a unified learning framework’, *Applied Intelligence* **4**, 297–316.

Studying Genotype-Phenotype Attack on k-anonymised Medical and Genomic Data

Muzammil M. Baig¹Jiuyong Li¹Jixue Liu¹Hua Wang²¹ School of Computer and Information Science

University of South Australia,

Mawson Lakes, South Australia 5095,

Email: {muzammil.baig jiuyong.li jixue.liu}@unisa.edu.au

² Department of Maths and Computing

University of Southern Queensland ,

Toowoomba, Queensland, 4350,

Email: wang@usq.edu.au

Abstract

Personal data of patients is largely collected at hospitals, clinics, labs etc. This data consists of medical and genomic record. Such patient data is shared for various health and research purposes. The utility of such sharing is worthwhile and its benefits are now well documented. It includes early diagnostic of some diseases like Phenylketonuria that can cause high chances of recovery. Population health analysis, derived from collaborative sharing of patient data, help government agencies to draft proper policies to raise the standard of living of people. On the other side of picture, many patients fear about the misuse of their personal data. This fear caused social (sometimes legal) requirement to properly safeguard the personal data before sharing. Various generalization techniques were suggested to anonymize the both types of patient data i.e. medical and genomic. Generalization based privacy protection technique, k-anonymity is considered to be one of important practices to anonymize the patient data. Due to rapid technological advancements, it is possible that the medical and genomic data of same patient(s) can be publically available from different sources. Such a scenario has created new privacy threats to patient data. Genotype-Phenotype attack is one of these threats. This research paper shows that how k-anonymised medical and genomic data is subject to genotype-phenotype attack.

Keywords: k-anonymity, genotype-phenotype attack, privacy compromise.

1 Introduction

Health organizations largely collect the personal information from patients. The range of this personal information has changed with time. Anciently it consists of few personal particulars like name, date of birth, address. Technological advancements added

new particulars and dimensions to patient personal information. Presently, patient information is divided into two major groups i.e. medical information and genomic information. Medical information is the set of those attributes that is directly asked from patient like name, date of birth, address, nationality etc. Moreover it also includes the medical state of patient like disease information and results of various laboratory test(s). Genomic information consists of DNA sequence of the patient. Healthcare organizations now can store gigantic medical and genomic information due to decreasing cost of archiving and DNA sequencing.

Large scale storing of medical and genomic information is very useful for research purposes. It provides bio-medical research community the opportunities that were severely limited or nonexistent before. The analysis of medical information through modern data mining techniques helps the researchers to come-up with interesting facts. It can lead to find out that which population is prone to which disease. The genomic data, also called DNA sequence, is the blueprint of species. The study of genomic data can help to find out the reasons behind analytical results of medical information. Genomic data is a precious resource for biomedical research. It helps medical researchers to understand causes of diseases and to find effective ways to cure the diseases. For example, the evolution of medical paradigms like personalized health care is a result of genomic analysis (M. West et al. 2006). Other examples include the discovery of individuals genotype influence on the metabolism of pharmaceuticals (Roses 2000, Burnett et al. 2003), and the formulation of new medicines based on subsequent results. Therefore, sharing medical and genomic data is crucial for modern biomedical research.

Patient medical and genomic data may be maliciously used while it is very helpful for biomedical research. Biological research has shown that the genomic data holds identity and sensitive information of an individual, such as gender and disease traits (Altschul 1990, Caenazzo 1998). Employers and insurance companies may utilize such private information for discrimination. For example, employers may refuse job to someone who has the trait of mental illness. Insurance companies may charge higher premium for some individuals. Due to this and many other reasons many people fear that information gleaned from their genomic data will be misused, abused, to influence their employment and insurance status, or simply cause social stigma (Clayton 2003, Xiao & Tao 2007, Hall & Rich 2000, Rothstein 1997). Genomic data, unlike most medical information, contains more specific information that can relate family

Table 1: Possible cases of 'Match Records'

Case #	Anonymised Medical Data				Anonymised Genomic Data		
	Quasi Identifiers			Sensitive Attributes	Quasi Identifiers		Sensitive Attributes
	Age	Sex	Zip	Phenotype	Age	Sex	Genotype
1	25 – 35	Male	500*	A	25 – 35	Male	A
2	25 – 35	Male	500*	B	35 – 40	Male	B
3	25 – 35	Male	500*	C	25 – 30	Male	C
4	25 – 35	Male	500*	D	40 – 45	Female	D

members too. Due to the sensitive nature of medical and genomic data there are many social pressures to protect the privacy of patients data.

Due to the serious nature of such issues, US Government adopted the Health Insurance Portability and Accountability Act (HIPA) (DHHS 2002) and Genetic Information Non-discrimination Act (GINA) to restrict the access of medical/genomic data by employers and insurance companies (GINA 2008). In the European Union, the Data Protection Act (DPA) of 1998 imparts strict regulations on the processing of personal data, of which genomic data is a part of (NHS 2000). So, if there is no proper assurance of privacy preservation, not only patients will be hesitant to provide but many data holders will be unwilling to share patient data. Such a situation can be harmful for new or continuing bio-medical research. In recognition of the problem, the protection of privacy for patient data is considered a major challenge for the biomedical research community (Altman & Klein 2002, Vaszar et al. 2003).

Realizing the need of privacy protection of patient information many researchers presented their research work to preserve the patients privacy both in medical and genomic data. Initially lot of research work purposed in the domain of medical data and lot of anonymization techniques were purposed like k-anonymity (Sweeney 2002), L-diversity (Machanavajjhala et al. 2007), m-Invariance (Xiao & Tao 2007), (α, k) (Wong et al. 2006). Subsequently ample developments in DNA sequencing caused ease accessibility to genomic databases. There are currently more than 100 genomic databases available online over the World Wide Web (Michael & Galperin 2007); most of them are freely available. Such huge availability of genomic data also raised the need of privacy protection. It motivated the researchers to purpose the techniques to preserve the privacy of genomic data like familial data protection (Gaudet et al. 1999), gene trustee (Burnett et al. 2003), generalization lattices (Malin 2005b).

Existing medical and genomic data anonymization techniques were designed without considering the availability of both types of data of the same patient. It is possible that two or more locations can publicly release the patient data of same patient (Malin & Sweeney 2004). Such situation caused genotype-phenotype attack, discussed in detail in section 3. Generalization based techniques like k-anonymity (Sweeney 2002) are largely discussed in literature. In this paper we proved that k-anonymization technique for patient data do not assure the privacy of patient against genotype-phenotype attack.

In rest of the paper, some related work in section 2. highlights the k-anonymity model for patient data publishing. Section 3. discusses genotype-phenotype attack and weaknesses of k-anonymity against this inference. Experimental results of genotype-phenotype attack on real work data are in section 3. Conclusion and possible future work is summed in section 5.

2 Related Work

Medical data was the first constituent with respect to the privacy of patient data. In the beginning, explicit identifying features of a patient, such as name, address, telephone number and social security number, were removed from the medical data before publishing. This practice was based on the assumption that anonymity is maintained because the resulting data look anonymous. Advocates of such practice claimed that such anonymization sufficiently protects the identity of the patient to whom the medical data corresponds. At a glance, such claims appear to be true. How can one learn the identity of such anonymised data, when there is no register linking anonymised data to identity?

But in her pioneer work Sweeney (Sweeney 2002) showed that 87% (216 million of 248 million) of the population in the United States had reported characteristics that likely made them unique based only on 5-digit ZIP, gender, date of birth. Further, by linking such anonymised medical data with voter registration list the medical record of governor of Massachusetts was successfully identified.

These analyses arose the need of some proper anonymization mechanism for the publishing of medical data. Sweeney also defined the k-anonymity modal in her pioneer work (Sweeney 2002). It states that a record in the quasi-identifier should be identical to at least (k-1) other records in a data set, and therefore, no individual is identifiable.

3 Genotype-Phenotype Attack on k-anonymity Tables

In this section we formally define genotype-phenotype attack. First we explain some preliminary definitions commonly used in k-anonymization.

3.1 Preliminary Definitions

A data set discussed in k-anonymization has the following attributes.

1. Identifier attribute (A_{ID}) is the attribute that identifies every record in medical data. An example of identifier attribute is social security numbers (SSNs) or names of patient.
2. Quasi-identifier (A_{QI}) attributes is a set of attributes ($A_{QI}^1, A_{QI}^2, \dots, A_{QI}^n$) in a table that potentially identifies individuals in the table. Normally, a quasi-identifier attribute set is specified by domain experts. For example zip code, age, sex of patient.
3. Sensitive attribute A_s is a set of attributes that contains private information of patients. It includes disease information etc.

Data sanitation before the publication will remove identifier attribute(s). However, the quasi-identifiers

Table 2: Medical and genomic tables before anonymization

	Identifier attribute	Quasi-identifiers			Sensitive attributes (Phenotype)		
Record #	Patient Name	Age	Gender	Zip	Disease	Disease State	Treatment
1	Alex	50	Male	5000	Refsum	4	Medicine – A
2	Adam	55	Male	5095	Huntington	3	Medicine – A
3	Noreen	40	Female	5001	Galactosemia	3	Medicine – C
4	Kathy	45	Female	5007	Fragile X	1	Medicine – F

(a) Patient Medical table

	Identifier Attribute	Quasi Identifiers		Sensitive Attribute
Record #	Patient Name	Gender	Age	DNA
1	Alex	Male	50	catg...
2	Richard	Male	47	actg...
3	Noreen	Female	40	ttag...
4	Tanya	Female	39	aacc...

(b) Patient Genomic table

can potentially reveal identity information of individuals. K-anonymization is used to generalize the values in the quasi-identifier attributes and make them indistinguishable. An equivalence class $E_c(A)$ with respect to an attribute set (A) is the set of all records in the table containing identical values for attribute set (A). In anonymization process, attribute set (A) refers to the quasi-identifiers. So proper notation of equivalence class is to be $E_c(A_{QI}^n)$. A table is k-anonymous with respect to quasi-identifiers if the size of equivalence class is ' k ' or more. The k-anonymity property can be defined as $\forall_i |E_i| \geq k$. K-anonymization is a process to modify a table to a view that satisfies the k-anonymity property with respect to the quasi-identifiers (A_{QI}^n). Where ' k ' indicates the strength of anonymization, larger the k , the stronger the privacy protection.

3.2 Genotype-Phenotype Attack

DNA sequences are very important biological information for studying human diseases. DNA sequences also contain rich privacy information of individuals. They can be used to identify not only a person but also whole family tree. Presently there is tremendous increase in the availability of DNA databases and more than 100 genomic databases are available online over the World Wide Web (Michael & Galperin 2007). Researchers proved that genomic data not only itself can exploit the privacy (Malin & Sweeney 2002) but it can also be linked with some other existing patient information to compromise the privacy of data holder (Malin 2000, 2001).

Clinical phenotypic abnormalities are the *phenotype* of a patient. For example, disease, disease stage, medication etc. Marker genes for some diseases and known hotspots of mutation and other characterized mutations in genes are the *genotype* of a patient. For example, some diseases (like Refsum and Galactosemia) can be identified from mutation in specific gene.

Based on biological knowledge, phenotype can be linked to genotype with certain precision. Such precision is improving with improved knowledge in genetic knowledge. Genes and gene mutations are linked to diseases or other medical conditions, so that DNA sequence is linked to medical record where patient is potentially identified from quasi-identifiers.

Genotype-phenotype attack already discussed by Malin (Malin 2005a). The DNA sequences are published with little personal identifiable information and are assumed secure for privacy. However, the

link of genotype-phenotype between medical and genomic records associates anonymous DNA record to a medical record, where quasi-identifiers contain rich personal identifiable information, and hence no more privacy security for DNA anymore. Genotype-phenotype attack is similar to the quasi-identifier based linking attack used in Sweeney's earlier work with health data re-identification (Sweeney 2000) on non-generalized data. If medical and gene data have not been anonymised, the chance of individuals being identified is great as discussed in (Malin 2000). In more practical scenario, quasi-identifiers of both medical and genomic data sets are k-anonymised before public release. When quasi-identifiers of both medical and genomic data sets are k-anonymised, will the genotype and phenotype attack exist? In the next subsections, we will mainly analyze the cause of the identification and demonstrate the risks in anonymised data.

Formally, we assume that there are two tables D_M and D_G . Where D_M contains patients' medical information including $\{Age, Sex, Zip, Medical\ conditions\ (phenotype)\}$. D_G contains patients' genome information including $\{Age, Sex, DNA\ sequence\ (genotype)\}$. Some records in both tables belong to the same group of patients, i.e. $D_M \cap D_G \neq \emptyset$. The genotype-phenotype attack is simulated as record match, which is defined as follows:

Definition 1: (Match) Match records (T_{Match}) is the pair of record that exists in both medical and genomic tables; when the genotype of a genomic record matches with phenotype of a medical record.

$$D_M(Phenotype) = D_G(Genotype)$$

One of the following situations occurs for each quasi-identifier attribute (A_{QI}^n) of the match records (T_{Match}) in both tables:

1. Values of quasi-identifier(s) of match records are same in both tables (medical and genomic).
2. Generalization ranges of quasi-identifier of match records of both tables (medical and genomic) overlap each other.
3. One of the generalized value interval of a quasi-identifier of matched records (either genomic or medical) is the sub-set of another one's.
4. Values of quasi-identifier do not match.

Table 3: Medical and genomic table publically released

Record #	Quasi-identifiers			Sensitive attributes (Phenotype)		
	Age	Gender	Zip	Disease	Disease State	Treatment
1	50 – 55	Male	50**	Refsum	4	Medicine – A
2	50 – 55	Male	50**	Huntington	3	Medicine – A
3	40 – 45	Female	500*	Galactosemia	3	Medicine – C
4	40 – 45	Female	500*	Fragile X	1	Medicine – F

(a) 2-anonymised Patient Medical table

Record #	Quasi Identifiers		Sensitive Attribute
	Gender	Age	DNA
1	Male	45 – 50	catg. . .
2	Male	45 – 50	actg. . .
3	Female	37 – 42	ttag. . .
4	Female	37 – 42	aacc. . .

(b) 2-anonymised Patient Genomic table

Table 4: Publically released Patient Genomic Record after genotype extraction

Record #	Quasi identifier		Sensitive attribute	Extracted Attributes (Genotype)		
	Gender	Age	DNA	Disease	Disease State	Medicine
1	Male	45 – 50	catg. . .	Refsum	4	Medicine – A
2	Male	45 – 50	actg. . .	Sickle Cell	1	Medicine – B
3	Female	37 – 42	ttag. . .	Galactosemia	3	Medicine – C
4	Female	37 – 42	aacc. . .	Refsum	3	Medicine – F

In the first three cases, privacy of the patient can be compromised. Now we formally define the privacy compromise.

Definition 2: (Privacy Compromise) if any equivalence class $E_c(A_{QI}^n)$ in k-anonymised table no more satisfies the k-anonymity property then the privacy of the records in that specific equivalence $E_c(A_{QI}^n)$ class is said to be compromised.

One trivial privacy compromise is that when both data sets have different k , after the matching, the anonymity level is reduced to a smaller k for the matching records. In this paper, we consider a non-trivial case that both data sets have the same k . Table 1. depicts the four possible cases of match records; first three can cause the privacy compromise.

In real world situation, the genotype-phenotype matching is more complicated. A genotype may not match a phenotype with 100% accuracy. There can be multiple matched and unmatched cases. One phenotype in the medical data set may match many genotypes in the genome data set, one genotype in the genomic data set may match many phenotypes in the medical data, or genotype and phenotype has many matches but with different patients. These cases will lead to a low successful rate for genotype-phenotype attack. However, privacy breaching is an individual event and one breaching may cause tremendous damage for a person. Therefore, average possibility is not applicable to privacy protection. Instead, lower bound should be considered in privacy protection. In the following discussions, we will show that there is real possibility that an individual can be identified by genotype-phenotype attack on the k-anonymised medical and genome tables.

Table 2. shows the sample medical and DNA record of different patients. We have a realistic assumption that both medical record and genomic data is held by two different locations. It is already proved (Malin & Sweeney 2004) that different locations can release the medical and genomic data of same patient(s).

Before the public release, both tables are k-anonymised. Table 3. depicts the publically released view of both tables. Both tables satisfy the k-anonymity property where $k = 2$.

Both data holders anonymize and release medical and genomic data independently. K-anonymity model is simple to understand and easy to implement. Now we apply genotype-phenotype attack on Table 3(a) and Table 3(b).

First, the genotype of patients is extracted from the DNA sequences of Table 3(b). Some traits like single gene diseases, disease state, medicine and dosage of medicine used can be extracted from DNA sequence. These extracted traits depict the ‘genotype’. Malin has already demonstrated there are minimum of 40 standardized diseases (via ICD-9 codes) to which DNA mutations in the genome are directly related (Malin 2000). These traits are the result of continuous efforts of bio-research community that are busy to find relationships between genomic mutations and the ability to process drugs and treatments (Altman & Klein 2002, Vaszar et al. 2003). These bio-research efforts also include activities like GWAS¹ and TCGA². Table 4. depicts the extended view of the patient genomic Table 3(b) after ‘genotype’ extraction.

Figure 1. illustrates the graphical representation of generalization hierarchies of Age attribute in both tables (medical and genomic).

With this information in hand, we can link the genotypes of genomic Table 4. with phenotypes of

¹Genome wide association study (GWAS) is a generic term, which National Institute of Health (NIH) (<http://nih.gov>) defines as including “any study of genetic variation across the entire human genome that is designed to identify genetic associations with observable traits (such as blood pressure or weight), or the presence or absence of a disease or condition.”

²The Cancer Genome Atlas (TCGA). A proposed project to chart the inherited and acquired mutations that relate to the onset, diagnosis, progression, and treatment of cancers, by genotyping bio-specimens and examining the genomic data in light of clinical data on the patients. Piloting is starting. (<http://cancergenome.nih.gov>)

Table 5: Anonymized tables after genotype-phenotype attack

		Quasi-identifier			Sensitive attributes (Phenotype)		
	Record #	Age	Gender	Zip	Disease	Disease State	Treatment
(T _{Match1})	1	50	Male	50**	Refsum	4	Medicine – A
	2	50 – 55	Male	50**	Huntington	3	Medicine – A
(T _{Match2})	3	40 – 42	Female	500*	Galactosemia	3	Medicine – C
	4	40 – 45	Female	500*	Fragile X	1	Medicine – F

(a) Patient Medical table after genotype-phenotype attack

	Record #	Quasi identifier		Sensitive attribute	Extracted Attributes (Genotype)		
		Gender	Age	DNA	Disease	Disease State	Medicine
(T _{Match1})	1	Male	50	catg...	Refsum	4	Medicine – A
	2	Male	45 – 50	actg...	Sickle Cell	1	Medicine – B
(T _{Match2})	3	Female	40 – 42	ttag...	Galactosemia	3	Medicine – C
	4	Female	37 – 42	aacc...	Refsum	3	Medicine – F

(b) Patient Genomic table after genotype-phenotype attack

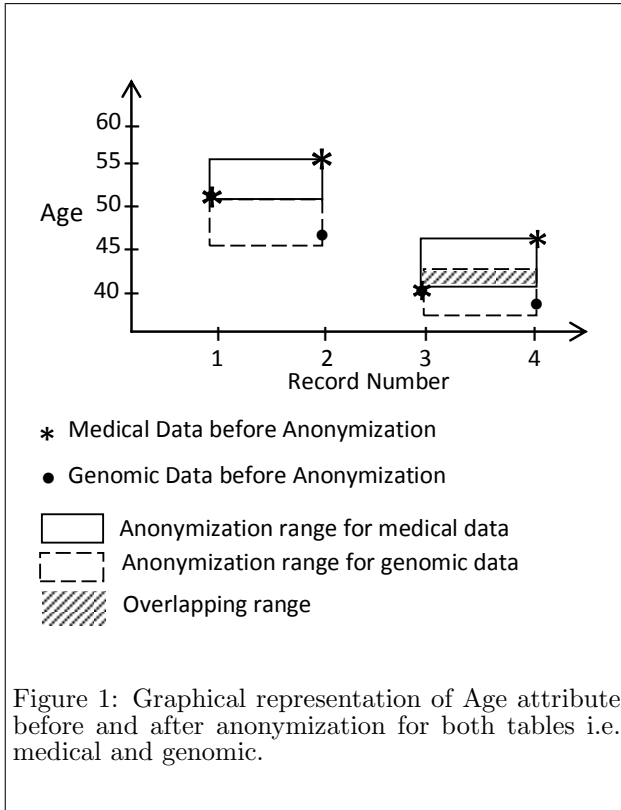


Figure 1: Graphical representation of Age attribute before and after anonymization for both tables i.e. medical and genomic.

medical Table 3(a). Record 1 and 3 of the genomic Table 4. are linked with record 1 and 3 of the k-anonymised medical Table 3(a); because the *genotype* (extracted attributes) of these genomic records match with the *phenotype* (sensitive attributes) of the corresponding k-anonymised medical records. As the anonymization range of *age* attribute in both tables is overlapping, so for linked records (1 and 3) the anonymization range of *age* attribute is also reduced to overlap part. Both anonymised tables (medical and genomic) are shown in Table 5(a),(b) respectively after genotype-phenotype attack.

It is possible that a genotype of genomic table may match with more than one phenotype in medical table or vice versa. Even in this case the privacy of patients is at risk because the k-anonymity property of both tables (medical and genomic) is compromised. Both medical and genomic Table 5(a),(b) are no more 2-anonymous. After genotype-phenotype attack, size of equivalence classes for both tables (medical and genomic) is reduced to 1, which is less than the original

value of k i.e. 2. As per the definition of k-anonymity property the size of equivalence class should be more than or equal to k . Both tables (medical and genomic) after genotype-phenotype are now also open to other re-identification risks (Sweeny 2002). Its clear that k-anonymity is prone to genotype-phenotype attack. Sharing the patient data is helpful in many ways but it is seriously considered that such sharing is only prospective if the privacy concerns of the data owners (like genotype-phenotype attack) are properly addressed.

4 Genotype-Phenotype Attack Experiments

In this section, we explain our genotype-phenotype attack experiments to illustrate the risk of this attack. First we define some of the definitions used in these experiments.

Definition 3: (True Match) A pair of match records (T_{Match}) , is said to be True Match $(T_{TrueMatch})$ when both records belongs to same person.

Definition 4: (False Match) A pair of match records (T_{Match}) , is said to be False Match $(T_{FalseMatch})$ when it do not belongs to same person.

True match is actual risk to privacy of patients. Situation of False Match is possible because one genotype/phenotype may match one-to-many records. For example, there can be two or more different patients having same age, gender and disease.

Definition 5: (True Match Precision) Ratio between true match records $(T_{TrueMatch})$ and Match (T_{Match}) records is the *True Match Precision* $P_{True}(GP)$ of genotype-phenotype attack. The True Match Precision $P_{True}(GP)$ can be defined as:

$$P_{True}(GP) = \frac{\sum T_{(TrueMatch)}}{\sum T_{(Match)}}$$

The maximum value of true match precision is 1, when all match records are the true match. Similarly when none of the match record is true match record the true match precision is 0. More the value of is closer to 1, higher the risk of privacy compromise.

In our experiments, we sample data from a survey data set (Asuncion & Newman 2007) to simulate medical and genome data sets. The survey data set has total 8 attributes. We use *Age* and *sex* as quasi-identifier for both data sets, and rest information is assumed as phenotype and genotype in simulated medical and genome data sets. Table 6. depicts the division of test data. There is a match if values apart from *Age* and *sex* in two records from medical

Table 6: Attributes of Test Data set

Attribute	A_{QI}/A_s
Age	A_{QI}
Work Class	A_s
Years of Education	A_s
Marital Status	A_s
Occupation	A_s
Race	A_s
Sex	A_{QI}
Native Country	A_s
Salary	A_s

Table 7: Division of Test Data set into two sub-data sets

	Simulated Medical	Simulated Genome
Initial Records	10000	10000
Overlap Records	6912	6912
Data set Size	16912	16912

and genome data sets are identical. We control the overlapping records in the both data sets.

Both quasi-identifiers (*age, sex*) present in medical and genomic data. Rest of (sensitive) attributes depict phenotype in Simulated Medical data set and genotype in Simulated Genomic data set. We used Mondrian (LeFevre et al. 2006) k-anonymization in these experiments. In all iterations of experiments both quasi-identifiers (age, sex) use the same generalization hierarchy. The initial characteristics of both data sets (simulated medical and genomic) are described in Table 7.

Adult dataset has 33824 distinct records. We divided distinct records into two datasets each of 16912 records, out of that 6912 records were marked as overlap records. We repeat five set of experiments, varying the ratio of true overlap records (from 0% to 100%) within overlap records. Table 8. depicts the ratio between true and false overlap records before each set of experiment.

In each set of experiment both data sets k-anonymised to different values of k then we calculated true and false match records for each k-anonymization. Table 9. depicts the results of our all set of experiments.

It is clear that in each set of experiments on average 90% true overlap records are correctly identified. All correctly identified true match records depict the success of genotype-phenotype attack and can breach the patients privacy.

Individual privacy breach is not the only risk of genotype-phenotype attack. As per the definition of privacy compromise, the privacy of the patients is on risk if anonymised table no more satisfies k-anonymity

Table 8: Division of overlap records between True and False overlap records for different experiments

Experiment #	Total True Match Records	Total False Match Records	% True Match Records
1	0	6912	0%
2	1728	5184	25%
3	3456	3456	50%
4	5184	1728	75%
5	6912	0	100%

property. In our experiments, one or both of the simulated data sets (Medical and Genomic) do not hold the k-anonymity property after genotype-phenotype attack. Figure 2. depicts the degree of k-anonymity before and after genotype-phenotype attack in simulated medical data set. To illustrate the actual privacy risk, fig 2. includes only those equivalence classes that contain the true match records.

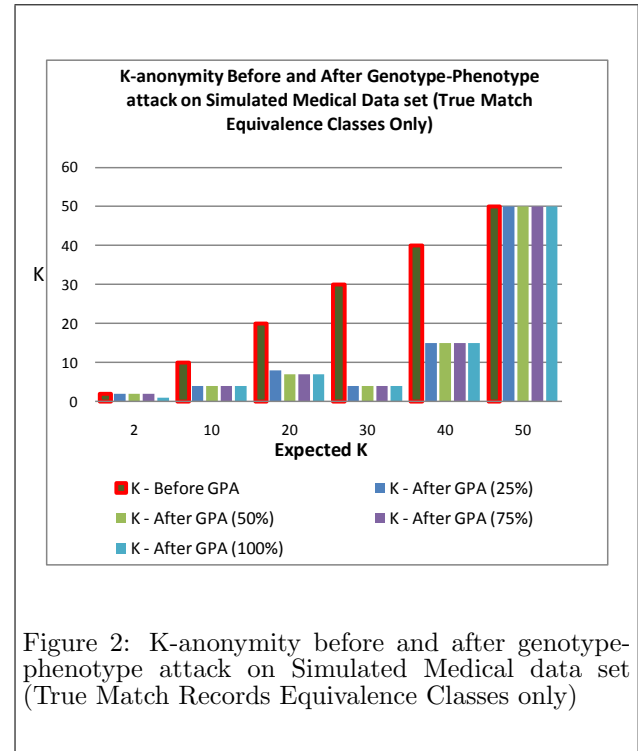


Figure 2: K-anonymity before and after genotype-phenotype attack on Simulated Medical data set (True Match Records Equivalence Classes only)

It is clear, that after genotype-phenotype attack either one or both data sets (Simulated Medical/Genomic) do not hold the k-anonymity property. In most cases when k-anonymity property is compromised, the degree of k-anonymity decreased to 50%. Interestingly, when $k=50$ the degree of k-anonymity remains unchanged after genotype-phenotype attack. For higher value of k , anonymised data set loses significant information and is of less beneficial. Any re-identification attack is only a risk when anonymised data is still useful.

5 Conclusion and Future Work

We have shown that genotype-phenotype attack is a risk to privacy preserving patient data publishing. We show through experiments that k-anonymity property can be compromised due to this attack. Data set compromised is subject to other re-identification. With the increased medical information available in medical records and better understandings of genetic links of diseases with gene defects, the threat of genotype-phenotype attack is increasing. Even though both medical and genomic data sets have been properly anonymised individually by different organizations, the change of genotype-phenotype attack still exists.

It is an important future work for privacy preserving data publishing to guard against genotype-phenotype attack.

References

Altman, R. & Klein, T. (2002), 'Challenges for biomedical informatics and pharmacogenomics', *Annual review of pharmacology and toxicology* **42**, 113–133.

Table 9: Test results of Simulated Medical and Simulated Genomic data sets

Total True Match Records	Total False Match Records	K	'K' after GP Attack (Simulated Medical)	'K' after GP Attack (Simulated Genomic)	Identified True Match Records	Identified False Match Records	True Match Precision
0	6912	2	1	1	0	5877	0
		10	4	1	0	5844	0
		20	7	20	0	5783	0
		30	4	30	0	5690	0
		40	12	40	0	5647	0
		50	50	50	0	5288	0
1728	5184	2	1	1	1502	4375	0.255
		10	4	1	1492	4352	0.255
		20	7	20	1474	4309	0.254
		30	4	30	1450	4240	0.253
		40	12	40	1435	4212	0.252
		50	50	50	1337	3951	0.253
3456	3456	2	1	1	2988	2889	0.508
		10	4	1	2968	2876	0.507
		20	7	20	2936	2847	0.506
		30	4	30	2891	2799	0.505
		40	12	40	2867	2780	0.504
		50	50	50	2665	2623	0.504
5184	1728	2	1	1	4457	1420	0.758
		10	4	1	4427	1417	0.756
		20	7	20	4377	1406	0.755
		30	4	30	4308	1382	0.752
		40	12	40	4275	1372	0.751
		50	50	50	3984	1304	0.753
6912	0	2	1	1	5877	0	0.999
		10	4	1	5844	0	0.998
		20	7	20	5783	0	0.997
		30	4	30	5690	0	0.993
		40	12	40	5647	0	0.992
		50	50	50	5288	0	1

Altschul, S. (1990), 'Basic local alignment search tool', *Journal of Molecular Biology* **215**(3), 403–410.

Asuncion & Newman (2007), 'Uci machine learning repository'.

URL: <http://archive.ics.uci.edu/ml/datasets.html>

Burnett, L., Barlow-Stewart, K., Pros, A. & H.Aizenberg (2003), 'The gene trustee: a universal identification system that ensures privacy and confidentiality for human genetic databases', *Journal of Law and Medicine* **10**(4), 506–513.

Caenazzo, L. (1998), 'Prenatal sexing and sex determination in infants with ambiguous genitalia by polymerase chain reaction', *Genetic Test* **1**(4), 289–291.

Clayton, E. (2003), 'Ethical, legal, and social implications of genomic medicine', *New England Journal of Medicine* **349**(6), 562–569.

DHHS (2002), 'Standards for privacy of individually identifiable health information, final rule', *Department of Health and Human Services, Federal Register* **67**(157), 53182–53273.

Gaudet, Arsnauld & Belanger (1999), 'Procedure to protect confidentiality of familial data in community genetics and genomics research', *Clinical Genetics* **55**(4), 259–264.

GINA (2008), 'Genetic information non-discrimination act (gina)'.

URL: <http://www.genome.gov/24519851>

Hall, M. & Rich, S. (2000), 'Patients' fear of genetic discrimination by health insurers: the impact of legal protections', *Genet Med* **2**, 214–221.

LeFevre, K., DeWitt, D. J. & Ramakrishnan, R. (2006), 'Mondrian multidimensional k-anonymity', *ICDE'06* p. 25.

Machanavajjhala, A., Kifer, D. & Gehrke, J. (2007), 'L-diversity: Privacy beyond k-anonymity', *ACM Transactions on Knowledge Discovery from Data* **1**(1).

Malin, B. A. (2000), 'Determining the identifiability of dna database entries', *Journal of the American Medical Informatics Association* pp. 537–541.

Malin, B. A. (2001), 'Re-identification of dna through an automated linkage process', *Journal of the American Medical Informatics Association* pp. 423–427.

Malin, B. A. (2005a), 'An evaluation of the current state of genomic data privacy protection technology and a roadmap for the future', *Journal of the American Medical Informatics Association* **12**, 28–34.

Malin, B. A. (2005b), 'Protecting genomic sequence anonymity with generalization lattices', *Methods of Information in Medicine* **44**(5), 687–692.

Malin, B. A. & Sweeney, L. (2004), 'How (not) to protect genomic data privacy in a distributed network: using trail re-identification to evaluate and design anonymity protection systems', *Journal of the American Medical Informatics Association* **37**, 179–192.

- Malin, B. A. & Sweeney, L. (2002), 'Inferring genotype from clinical phenotype through a knowledge based algorithm', *Proceedings of the Pacific Symposium of Biocomputing* pp. 41–52.
- Michael, Y. & Galperin (2007), 'The molecular biology database collection', *Nucleic Acids Research* **35**(D3-D4).
- M. West, Ginsburg, G., Huang, A. & Nevins, J. (2006), 'Embracing the complexity of genomic data for personalized medicine', *Genome Research* **16**, 559–566.
- NHS (2000), 'Data protection act 1998 - protection and use of patient information', *NHS Executive, (HSC 2000/009)*.
- Roses, A. (2000), 'Pharmacogenetics and pharmacogenomics in the discovery and development of medicines', *Nature* **38**, 815–818.
- Rothstein, M. (1997), *Genetic Secrets: Protecting Privacy and Confidentiality in the Genetic Era*, New Haven: Yale University Press.
- Sweeney, L. (2000), Uniqueness of simple demographics in the u.s. population, Technical Report LIDAP-WP4, Data Privacy Laboratory, Carnegie Mellon University, Pittsburgh, PA.
- Sweeney, L. (2002), 'K-anonymity: a model for protecting privacy', *International Journal on Uncertainty, Fuzziness, and Knowledge-based Systems* **10(5)**, 571–588.
- Vaszar, L., Cho, M. & Raffin, T. (2003), 'Privacy issues in personalized medicine', *Pharmacogenomics* **4(2)**, 107–112.
- Wong, R. C.-W., Li, J., Fu, A. W.-C. & Wang, K. (2006), '(alpha, k) anonymity: An enhanced k-anonymity model for privacy preserving data publishing', *KDD06* pp. 754–759.
- Xiao, X. & Tao, Y. (2007), 'm-invariance: Towards privacy preserving re-publication of dynamic data sets', *SIGMOD* pp. 698–700.

Building a Generic Graph-based Descriptor Set for use in Drug Discovery

Phillip Lock¹Nicolas Le Mercier²Jiuyong Li¹Markus Stumptner¹

¹ Advanced Computing Research Centre, University of South Australia,
Mawson Lakes, South Australia 5095,
Email: phillip.lock@unisa.edu.au

² Ecole Nationale Supérieure de Techniques Avancées, Paris, France
Email: lemercier@ensta.fr

Abstract

The ability to predict drug activity from molecular structure is an important field of research both in academia and in the pharmaceutical industry. Raw 3D structure data is not in a form suitable for identifying properties using machine learning so it must be reconfigured into descriptor sets that continue to encapsulate important structural properties of the molecule. In this study, a large number of small molecule structures, obtained from publicly available databases, was used to generate a set of molecular descriptors that can be used with machine learning to predict drug activity. The descriptors were for the most part simple graph strings representing chains of connected atoms. Atom counts averaging seventy, using a dataset of just over one million molecules, resulted in a very large set of simple graph strings of lengths two to twelve atoms. Elimination of duplicates, reverse strings and feature reduction techniques were applied to reduce the path count to about three thousand which was viable for machine learning. Training data from twenty six data sets was used to build a decision tree classifier using J48 and Random Forest. Forty three thousand molecules from the NCI HIV dataset were used with the descriptor set to generate decision tree models with good accuracy. A similar algorithm was used to extract ring structures in the molecules. Inclusion of thirteen ring structure descriptors increased the accuracy of prediction.

Keywords: Drug Discovery, QSAR, Molecular Graphs, Simple paths, Machine Learning.

1 Introduction

Drug molecules moderate metabolic activity by binding with the proteins (enzymes) engaged in metabolism. The binding activity with a particular protein depends primarily on the complementary shapes of the ligand (drug) and receptor site on the protein surface (Rush et al 2005), (Schnecke et al 2006). The chemical properties of the molecule at the bonding site are also important, but to a lesser extent. There is a strong commercial and human health interest in being able to predict how well a particular molecule will bind to a particular protein, but the molecule's 3D structure alone is not in a form that is suitable for comparison and prediction. The prediction of molecular activity from its 3D structure

is called QSAR (Quantitative Structure-Activity Relations) and there are many techniques in this field (Kubinyi 1997). In every case the 3D structure data must first be transformed into a descriptor set (Xue et al 2000) to which machine learning can be applied.

Descriptor sets used in QSAR are of three types, 1D, 2D and 3D each of which can be numerical or categorical. The first type includes sets which indicate the presence or absence of a chemical property or feature with a binary digit. A long string of binary digits forms a 1D chemical fingerprint of the molecule. The numerical version of a 1D descriptor set consists of a list of quantitative properties, such as boiling point, molecular weight, refractive index, density and logP (logarithm of the octanol-water partition coefficient). The 2D descriptors are those that are derived from connectivity data while the 3D descriptors are those that encapsulate information derived from spatial properties such as electrostatic and steric fields within the molecule. Techniques such as CoMFA and CoMSIA (Kubinyi 1998) are not only successful at prediction but because each descriptor is related to a spatial point in a molecule they can also provide information about which parts of a molecule are involved in bonding to a particular protein. Many of the details of QSAR techniques have been incorporated into commercial software packages and have had limited exposition in the literature.

This study reports on a project to build a 2D descriptor set based on simple paths extracted from molecular graphs. The intention has been to define a dictionary of paths that is widely applicable and useful for categorizing molecules and predicting their properties. At this time there does not appear to be such a set available in the public domain.

1.1 Outline of the paper

In Section 2 the sources of molecular structure data are listed and described. Section 3 introduces molecular graphs, paths, the generation of simple paths and the filtering of the resulting path sets. Here we describe the methods of feature selection which were used to reduce the paths dataset to a manageable size. This section also describes the algorithm used to construct a 1D descriptor for each molecule based on the path sets. In Section 4 we describe the categorization of the training sets and discuss issues of category imbalance. We outline the methods we used to adjust the balance of categories in the training data. Section 5 introduces the machine learning algorithms used in this study: Decision Trees and Random Forest. We also address the important issue of evaluating the outputs of these algorithms. In Section 6 the experiments are described and their results are displayed and discussed. Section 7 reports on the addition of chemical ring descriptors to the path sets.

Copyright ©2009, Australian Computer Society, Inc. This paper appeared at the Eighth Australasian Data Mining Conference (AusDM 2009), Melbourne, Australia. Conferences in Research and Practice in Information Technology, Vol. 101. Paul J. Kennedy, Kok-Leong Ong, and Peter Christen, Eds. Reproduction for academic, not-for profit purposes permitted provided this text is included.

We describe the method used to extract 13 different chemical ring structures and the effect on algorithm performance when they are included in the descriptor set. Section 8 reports on an additional application of the descriptor set to the classification of molecules into their chemical families. Finally Section 9 draws some conclusions and discusses possibilities for future work.

2 Data sources

Molecular structure data from a number of sources was used in this study (refer to Table 1). Initially the Ligand Info subset (von Grotthuss et al 2004) was used for generating a set of simple paths. The NCI dataset (NCI data 2004) contains HIV activity data which allowed it to be used as training data once a descriptor set had been created. In addition 26 sets with structures and numerical activities were obtained. These were collected from the literature (Mittal et al 2008) and, for convenience will be referred to as the 'literature sets' labelled 'LitSet'.

Dataset	Description	Size
LI NCI	LigandInfo Subset NCI HIV dataset	1.1 million 43,211
	Literature Sets (LitSet)	
ACE	Inhibitors of angiotensin converting enzyme	114
ACHE	Inhibitors of acetyl-cholinesterase	111
BZR	Inhibitors of benzodiazepine receptor	163
COX2	Inhibitors of cyclooxygenase-2	322
DAT	piperidine analogues for dopamine transporter	42
DHFR	Inhibitors of rat dihydrofolate reductase	397
ECR	binding of diacylhydrazine to ecdysone receptor	53
GPB	Inhibitors of glycogen phosphorylase b	66
GSK3B	Binding to Glycogen synthase kinase 3 beta	42
THERM	Inhibitors of thermolysin	76
THR	Inhibitors of thrombin	88
COMT	Inhibitors of catechol-O-methyltransferase	92
MX	Mutagenicity of mutagen X analogues	29
DR	Antagonists of dopamine receptor	38
GHS	Growth hormone secretagogue mimics	31
YOPH	Inhibitors of Yersinia protein tyrosine phosphatase	39
STRDS	Binding of steroids to carrier proteins	21
PTC	Phase-transfer asymmetric catalysts	40
RYSR	Binding of ryanoids to the ryanodine receptor	18
HIVRT	Inhibition of HIV-1 reverse transcriptase	101
ARB	Non-peptide angiotensin II receptor antagonists	28
KOA	Kappa opioid antagonists	39
TCHK	Inhibition of Trypanosoma cruzi hexokinase	42
ERB	Estrogen receptor binders	123
CBRA	Cannabinoid CB1 receptor agonists	32
ATA	Anti-tuberculosis agents	72

Table 1: Data sources

3 Molecular graphs

Because a molecule consists of atoms connected by bonds it can be naturally represented by a mathematical structure called a graph. A graph consists of a set of nodes connected by a series of edges. In this case the nodes are atoms and the edges are bonds. Because molecules can include ring structures the graphs can be cyclic.

3.1 Simple paths

A simple path is a list of nodes that are connected in sequence but with the constraint that a node cannot be repeated. A simple path, extracted from a molecular graph, will not revisit the same atom as it traverses ring structures, as shown in Figure 1. The molecules of interest in this project ranged in molecular weight from 20 to 300 atoms, with most including several ring structures. As a dataset of over one million molecules was used, the number of simple paths of various lengths that can be extracted soon becomes extremely large.

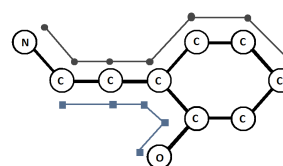


Figure 1: Examples of 2 Simple paths extracted from a molecular fragment. Here CCCCCO and NCCCCC

A number of algorithms for extracting simple paths from a graphs are available (Danielson 1968), (Bezemer et al 1987). In this project, an enhanced version of the depth-first algorithm was used to maximize performance and keep memory requirements in reasonable bounds. Extracting all possible paths from a typical drug molecule (70 non-hydrogen atoms) required about 50 ms. For reasons of computational efficiency the path lengths were restricted to the range 2 to 12 atoms, reducing the processing time to 5ms per molecule. A computer with an Intel Core2Duo, 2.40 GHz processor and 4 Gb RAM, running Java 1.6 on MS Vista, was used.

The path algorithm implemented filters that discard inverse strings (CCNO = ONCC), duplicates and those containing exotic atoms. The atoms retained are shown in figure 2.

Filters	Number of simple paths(2-12) remaining after filtering
No filtering	2.114 billion
Inverses	1.057 billion
Duplicates	423,504
Atoms	182,847

Table 2: Number of paths extracted

The LigandInfo Subset includes molecules with atoms that are unusual in biological systems. For example, exotic atoms such as radioactive Europium were present. A frequency distribution of atoms in the data was extracted and atoms with low frequency of occurrence were excluded. Paths containing these atoms were not extracted. Hydrogen atoms, although common, can only occupy ends of path strings as they form only one bond. Typically these are excluded from descriptors sets in QSAR (Xu et al 2000)

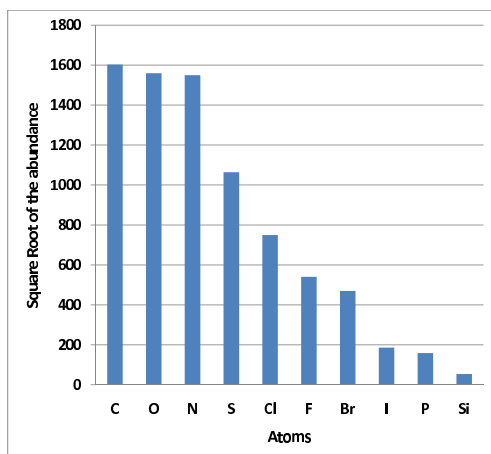


Figure 2: Atomic frequencies

3.2 Feature selection

3.2.1 Minimum redundancy

The remaining set of 182,000 paths is still too large for building a descriptor set of a practical size. Feature Selection is a technique commonly used in machine learning for selecting a subset of relevant features for building robust learning models. By removing irrelevant and redundant features from the data, feature selection helps improve the performance of learning models. In this project, two types of feature selection were employed: minimum redundancy selection and apriori selection.

Minimum redundancy selection is a process of removing every feature whose information content is already fully encapsulated within another feature. This process is similar to the computation of minimal generators (Nehmé et al 2005). In the set of 182,000 paths, it was evident that many paths were present in the same number of molecules as some of the sub paths within them. For instance: N-O-N appeared in 2157 molecules, C-N-O-N appeared in 2157 molecules and C-C-N-O-N also appeared in 2157 molecules. For the path CCNON to appear, CNON and NON also have to be present. As all are present with the same frequency, any one of the three implies the existence of the other two: there is redundancy. In feature selection the simplest information is typically kept, as the shortest instance carries as much information as the longer ones. However, from the chemists' point of view, the longer the path string, the more interpretable it is: knowing that CCNON exists implies more about the spatial and chemical properties of a molecule than knowing that a path NON is present. It was decided to keep only the longest redundant path string (in the example: CCNON) and to discard the shorter ones (here: NON and CNON) as it would be more meaningful for researchers using these descriptors. Once redundancies had been removed the path set reduced from 182,000 to 123,342 paths.

3.2.2 Apriori

Further feature reduction was attempted using the apriori algorithm (Jovanoski et al 2001). This method finds subsets which are common to at least a minimum number C of the given itemsets. This number is called the minimum support. It is asserted that when a subset is not sufficiently common, it is less likely to be a relevant feature. In this study, a subset is a path, and the number of itemsets where it can be found is its frequency in the molecules dataset. The advantage of this technique is that the support value

C can be set to retain only a reasonable number of paths. However, the apriori algorithm can reduce the path set to the point where a number of molecules are left with identical path descriptors: those that remain are incapable of being used to distinguish between these molecules. It is inappropriate to remove some of these rare but important paths because of their discriminating power.

A modified version of the apriori algorithm was implemented that uses backfilling to restore necessary features that would otherwise have been removed. If one path needs to be retained for its descriptive power, it is added back to the set of paths, even if its frequency is under the limit C . However, this requires many comparisons at each step of the algorithm to see if any molecule is no longer described and therefore needs backfilling. This adds processing time and also requires a lot of memory to store the molecular descriptions for comparison. Applying this algorithm to the LigandInfo Subset data, it was estimated that over 670 billion comparisons would need to be made. With the resources available this was not possible. The alternatives are to use a different algorithm or work with a smaller data set: both options were explored.

3.2.3 A Heuristic feature reduction method

Faced with limits on the dataset size that could be practically reduced using the apriori algorithm, a method that relied more on knowledge of the data and judgment was applied. A set of descriptors have been created from the 123,000 paths distilled from the LigandInfo database after minimum redundancy selection had been used. The heuristic method discards paths that are present in most molecules and therefore of less discriminating power, and also those that are too infrequently present to be universally useful. Once a core of paths was selected it was enhanced by backfilling: adding back in paths that had been discarded in the initial cut but were found to be necessary to represent particular molecules that would otherwise have insufficient descriptors for effective data mining. This method is much less demanding of computing resources and was able to operate within the available computing environment. The selection of the core set was guided by a graph, Figure 3, that shows the number of paths plotted against the frequency of their occurrence in the data set. Rare paths are those that are found in few molecules. Confusingly, there is a large number of individual paths that are rare. Common paths are those that are found in many molecules but these individual paths are less numerous. This shows that a small percentage of molecules contain a large number of the 123,000 available paths and a larger percentage of molecules have a smaller number of the paths. Only 5% of the paths were present in more than 2000 different molecules while 25% of the paths were found only once in the data set of molecules.

A number of different core sets were arbitrarily chosen from our set of 123,000 paths to determine which set generated the most useful descriptors. Four groups of paths each containing paths centred on a particular frequency in the data set were chosen. These were labelled G5, G10, G15 and G20 respectively. The numerical part in G20 indicates that any particular path from the core was represented in 20% of the molecules in the data set. Most molecules were still well represented by different paths from this core. However, this arbitrary selection is unable to guarantee that all molecules will be adequately represented by the limited set of paths. Backfilling was used in these cases to repopulate the core set with additional paths representing these molecules. Backfilling was

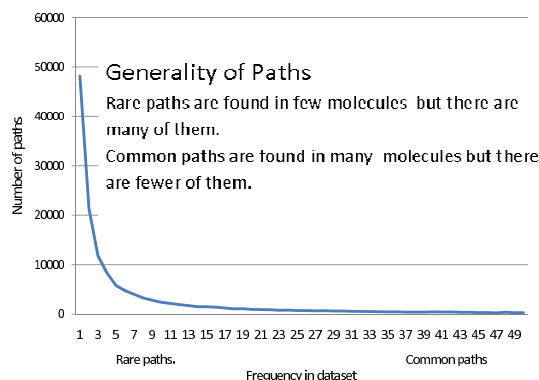


Figure 3: Path frequencies

continued until each molecule was represented by at least 10 paths. The initial frequency of the paths in the core was important: if it was too high, the paths in the set are too common and thus less expressive of differences in molecules. If the chosen frequency was too low, too many paths would need to be back-filled to compensate for the initial poor descriptive ability and would result in large path sets, possibly too large for machine learning. The initial frequencies were chosen at 5%, 10%, 15% and 20% of all the molecules. Each core set started with 3700 paths (3% of the 123,000 paths): each set had the same initial size, but with different paths. Tests were done on sets of 2,000, 10,000 and 50,000 molecules to ensure that the growths of the core sets was not excessive. The selection of the initial percentage, the range of paths to be included initially, and the minimum number of paths required for each molecule was made by trial and error. The alternative of optimising these selections formally was judged to be too computationally intensive. This remains an interesting area for future investigation.

Core Group	Centre Frequency	Final group size
G05	5%	4706
G10	10%	4533
G15	15%	4434
G20	20%	4360

Table 3: Final reduced path-set sizes

The resulting sets, displayed in Table 3, each had a reasonable size suited for machine learning algorithms. The growth in each path set size is a result of backfilling. The same method was tried starting with a single path rather than a core, and relying entirely on backfilling for populating it, but the results were poor.

The problem now remained of selecting the most descriptive of the four path sets. In order to determine which of the sets contained information about the largest number of molecules in the dataset, the paths extracted from each molecule were compared with those available in the reduced path sets. The percentage of found paths was plotted against the number of molecules in fig 4.

For each molecule, the ratio of each set was calculated: the number of paths present in both the molecule and path set divided by the total number of paths in the molecule. For instance, if a molecule has 50 extractable paths but only 20 of them are in the path set, then the ratio of this specific molecule for this specific set is $20/50=0.4$ or 40%. A ratio of 0 means that the molecule is not described at all. A ratio of 100% means that the molecule is fully de-

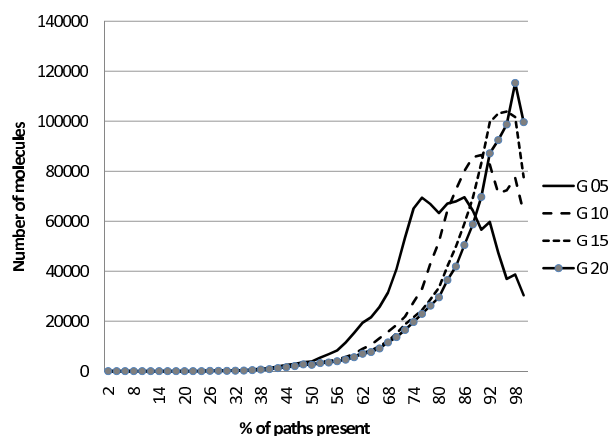


Figure 4: Number of molecules vs % of paths found in path set.

scribed. The G20 path set most closely approaches this so it was selected as the preferred descriptor set.

3.2.4 Building a path set with a smaller molecular data set.

In section 3.2.2 it was stated that the apriori algorithm with backfilling was unable to be used with the large data set derived from the LigandInfo data. The option of using a different feature reduction algorithm was explored in section 3.2.3. The alternative of using a smaller data set with apriori was also undertaken. The data used was taken from the LitSet in Table 1, 2240 molecules in total. After extracting paths and removing redundancies, 5594 paths remained. The apriori algorithm with backfilling and minimum support of $C=9$ was applied which resulted in a final path set of 2470 paths. This path set was named MS9 (minimum support 9). The paths in common between the G20 and the MS9 path sets are shown in Fig 5. The G20 path set shared 40% of its paths with the MS9 path set.

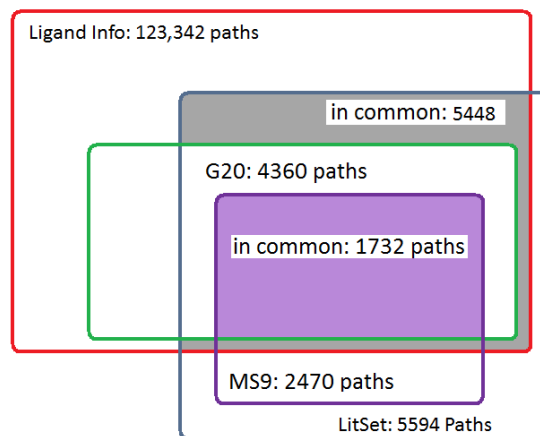


Figure 5: Paths in common between the path sets.

3.3 Descriptor preparation

Having settled on G20 (4360 paths) and MS9 (2470 paths) as the most promising path sets, the task remained to use them to build descriptor sets for individual molecules. In order to provide a descriptor set that is consistent between molecules, the paths were

arbitrarily assigned an order number. Paths strings were first ordered by length then by alphabetical order. A sample descriptor was then built by extracting all paths of length 2 to 12 from a sample molecule. At the same time the descriptor path set was scanned to see whether each path in the molecule is represented there. Where a path is found in both the molecule and in the path set, a '1' character is appended to the descriptor string; when a path from the descriptor set is not found in the sample molecule a '0' is appended. Refer to fig.6. Paths in the molecule that did not exist in the descriptor set were ignored. In the case of the G20 path set, this results in a binary string of length 4360, with MS9 the descriptor string length is 2470. Note that multiple occurrences of the same path in a molecule are not taken into account.

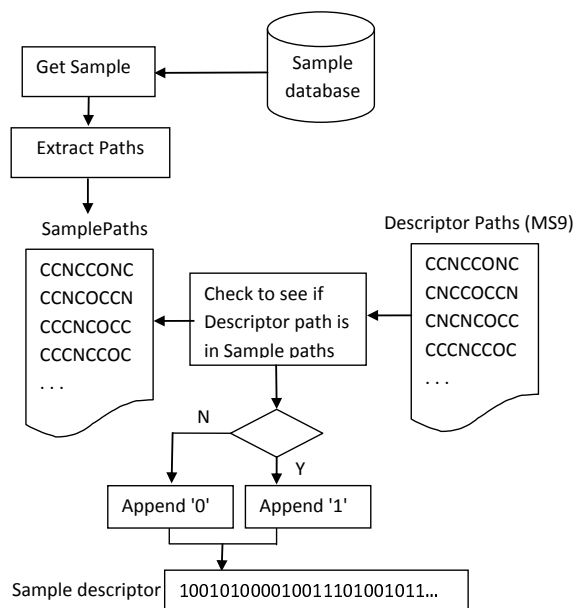


Figure 6: Flowchart showing the building of a descriptor for a sample.

4 Categorization of training sets

Seven of the data sets with more substantial molecule counts were taken from the 'literature sets' of molecules for training and testing as was the NCI HIV data set of 43,211 molecules. Both sets are provided with activity scores, in the first case with numerical values, in the second the data was annotated as Inactive (CI), Moderately active (CM) and Active (CA). In the case where the activities are numeric, they were first converted to categories. The decision was made to divide the scores into two categories, Active and Inactive. After testing it was decided to set the boundary between active and inactive at a level that categorised molecules with activities below 75% of max activity range as inactive. Those in the top 25% were initially categorised as active. All molecules with activities within 5% of the boundary were removed to demarcate more clearly between actives and inactives.

This results in molecules below 70% being defined as inactive and those in the top 20% as active. Refer to fig.7. This imposes some imbalance in the data, but this is difficult to allow for because of the distribution of activities within the molecule data sets. The HIV dataset of 43,211 molecules is already categorised but the data is massively imbalanced. There are 41,625 inactives, 1114 moderately actives and 472 actives. Imbalanced data makes the prediction accuracy misleading. The actives represent only 1% of the

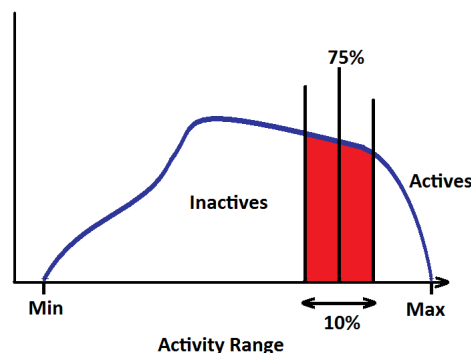


Figure 7: Setting the category boundary

whole data set so a crude model such as 'all samples are inactive' will be correct 99% of the time. To counter this, the data was under-sampled by repeatedly taking a random selection from the inactives of the same size as the actives (Drummond et al 2003). This produced many separate balanced data sets for use in model building.

5 Machine Learning

The nature of the descriptors is categorical which favours a classification algorithm such as a Decision Tree or Random Forest algorithms. These two algorithms were applied to the training sets. Both algorithms used were from the Weka suite of data mining tools (Frank et al 2004).

5.1 Decision Trees

The decision tree algorithm (Kohavi et al 2002) classifies samples into categories, in this case active or inactive. In building a tree model sets of the values, for a particular descriptor across the set of samples, are inspected in turn to identify the descriptor whose values can be split into two groups which have the greatest purity. Purity is measured by information gain or by the Gini coefficient. In Weka's J48 implementation the information gain is used. Having identified the most productive descriptor for the first split, the process is repeated recursively on samples from each half of the split to locate the next most useful descriptor on which to split. This process continues until the remaining sets each contain one sample.

The tree that results embodies a model for prediction but it is likely to overfit as it is closely tied to the training data and it may not perform well with general sample data. Pruning removes subtrees by replacing a node with a leaf when that improves the estimated error with generalised test data: it makes the tree model more general.

5.2 Random Forest

Overfitting can be reduced by using the Random Forest (Breiman 2001) meta-learner algorithm. As its name suggests, a random forest is built using many trees. Given a set of samples, each with a number of descriptors, many training sets are built using a random subset of the descriptors. Each subset is randomly extracted from the full descriptor list each time: used descriptors are replaced. A tree is built for each subset with the descriptor splits and split values that occur most frequently being used for the final model. Because the final model is made from many trees each based on a subset of the sample descriptors,

it is less strongly dependent on particular descriptors and therefore less prone to overfitting.

5.3 Evaluation of models

The classifier algorithms build tree models and report the accuracy of the models. The performance is summarised in the confusion matrix. In this case where there were only two categories the results were displayed as four numbers, the numbers of True Positives (TP), False Positives (FP), True Negatives (TN) and False Negatives (FN). The TP and TN scores together form the total of successful predictions.

	Predicted Class		
Actual Class		+ve	-ve
	+ve	TP	FN
	-ve	FP	TN

Table 4: Confusion Matrix for a 2-Class Problem

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} * 100$$

The accuracy is defined as the percentage of successful predictions out of the total number of samples classified. However, the accuracy measure alone is unreliable in quantifying the performance of a model. For example, a binary classification of balanced active and inactive samples will produce an accuracy of 50% with a random classification of each sample. The apparent quality of a model may be due more to the distribution of categories in the data than to the model itself. A method of evaluating models is needed.

The ROC (Receiver Operating Characteristic) curve is a graph of the True Positive Rate (*TPR*) vs. False Positive Rate (*FPR*) of a model (Fawcett 2007). The Area Under the Curve (*AUC*) can be used to evaluate the quality of a classification. This tool helps to compare different models independently of the class distribution. Therefore AUC can be used with decision trees (Ferri et al 2002) and provide more information than the accuracy measure which is affected by category imbalance in the data (Huang et al 2005). A perfect model would have a score of 1 and a random draw model would have an AUC of 0.5. The higher the value above 0.5, the better the classifier.

6 Experiment and results

6.1 Using the literature sets

The activities of LitSet sets with more than 100 samples (ace, ache, bzt, cox2, dhfr, erb, hivrt) were divided into two categories as described in Section 4. This was restricted to the larger sets as the smaller sets would be more prone to overfitting. The classification of the molecules was done using the J48 decision tree and Random Forest algorithms. In all cases 10 fold cross-validation was applied. For each set of molecules, the accuracy and the AUC of the models generated by both algorithms was tabulated, as shown in Figure 8 and 9.

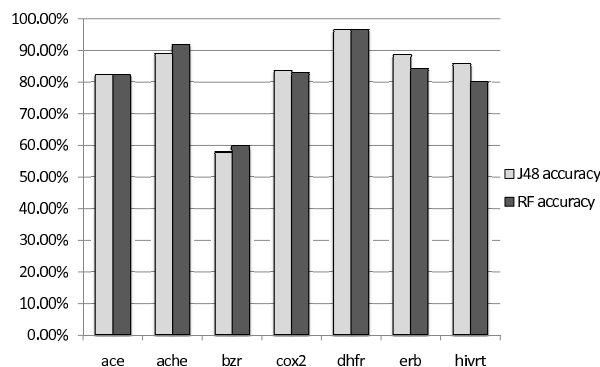


Figure 8: Accuracy of the models generated by J48 and Random Forest (RF) using the MS9 descriptors

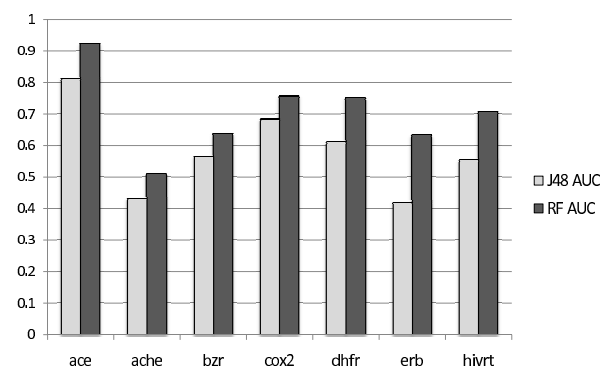


Figure 9: AUC of the models generated by J48 and Random Forest (RF) using the MS9 descriptors

Except for the *bzt* set, all sets were predicted with an accuracy above 80% when using the J48 or Random Forest algorithm. However, J48 produces models with a poor AUC, sometimes below 0.5 (equivalent to a model with worse predictive power than a random draw) showing that there was overfitting and then pruning. Random Forest generates models with better AUC as it tends to eliminate overfitting. Those results are comparable to other studies in the literature. Finally, the G20 set of paths gives results equivalent to the MS9 (with a difference of 1-2% for the accuracy and difference up to 0.3 for the AUC).

The two sets of paths give mixed results when classifying the molecule sets from which they were generated. The accuracy of the models generated by random forest and J48 are promising, but the AUC scores are quite low for J48: the efficiency of the generated models clearly depends on the machine learning algorithm used.

6.2 Using the NCI HIV database

As inactive molecules in the NCI HIV are far more numerous than the active ones, the sets were balanced by selecting a matching number of inactives and actives. The 472 actives and 472 randomly chosen inactives were compared 50 times to obtain average results. Table 5 display the results of the models generated using J48 and the path set MS9.

The standard deviation, minimum and maximum scores reveal that all 50 tests performed equally well. The average accuracy and AUC are comparable to other studies in literature. Similar results were found for LitSet activity classification. As the MS9 set of path was generated from the LitSet, good results with the LitSet were expected. However, the NCI HIV

	Accuracy	AUC
Average	83.52%	0.845
Std. deviation	1.36%	0.015
Minimum	80.56%	0.811
Maximum	87.34%	0.878

Table 5: Classification of inactive and active molecules of the NCI HIV database

molecules had no relation to the MS9 subset and they were still classified accurately. The conclusion can be drawn that the MS9 set of paths forms the basis of a descriptor set with good general predicting power even when applied to molecules not directly involved in its generation.

7 Enhancing the descriptor set

The algorithm to extract simple paths is closely related to algorithms that select simple circuits. A simple circuit in a cyclic graph is a ring path which excludes the repeated node. In a molecular graph this is equivalent to extracting chemical ring structures. A modified version of the path algorithm was used

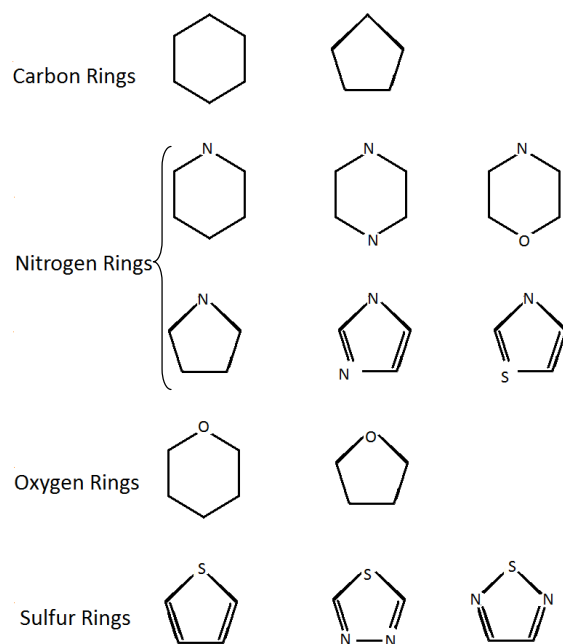


Figure 10: Rings detected

to detect 13 of the most common molecular rings: 2 C-Hetero rings, 6 N-Hetero rings, 2 O-Hetero rings and 3 S-Hetero rings (Xu et al 2000), (Hanser et al 1996). Refer to fig.10. The ring detection was performed independently of the path detection, to allow the possibility of using either or both paths and rings to build a descriptor. The new algorithm just had to detect a path corresponding to one of the unfolded rings and then discover the difference between a true ring and a regular path by checking if it was possible to close the ring. This detection takes less than one millisecond per molecule compared to 2 to 10 millisecond for the path detection. Thus it does not affect the general execution time of the algorithms. Knowing that many molecular families have a char-

acteristic number of rings (for instance the steroids have 3 rings with 6 C and one with 5 C) the number of rings in each molecule rather than just their presence was used. This additional feature never diminished the previous results and even improved some: the molecule classification into chemical families, see below, improved by 1% (to 94.6% from 93.5% without ring detection), and the active/inactive classification improvement ranged up to 5.1% (bzz set with J48). Note that the AUC tends to remain the same whether ring detection is used or not.

8 Classification of molecules into chemical families

A test was run with the MS9 set of paths on the 26 sets of the LitSet to see if a model was able to classify molecules into their correct chemical groups. Using J48, an accuracy of 93.5% was achieved: almost 2100 molecules out of 2240 had their group correctly predicted. The accuracy of this model increased to 94.6% when ring descriptors were included. With ring detection, G20 performed equally well as MS9 with 94.56% accuracy.

When all the molecules, classified as active in the NCI HIV database, were included, the accuracy of classification into 26 plus 1 (HIV) groups was 92%: 87% of the NCI HIV molecules were correctly classified.

The first test, with 26 sets, reveals that the reduced path sets contain sufficient information to distinguish between several types of molecules. Even though the 3D structural information was removed, the specific signature of many types of molecules remains encapsulated in the retained paths. The second test, using the additional HIV set, also confirms that these sets of paths are able to perform well, even with molecules not used in creating those sets. This indicates that the MS9 and G20 path sets are probably widely usable in the drug discovery domain.

9 Conclusions and further work

This study has identified two sets of simple paths, of length 2 to 12 atoms, that have been used to generate successful molecular descriptor sets. These path sets were tested with a variety of molecule data sets and were able to achieve an average accuracy score of 83% for all sets. The accuracy score was supported with an AUC measure of 0.71 with a mixed set of 2000 molecules and an AUC of 0.85 with 43,000 molecules from the NCI HIV dataset.

These descriptors were also found to be efficient at classifying individual molecules into chemical families. This worked with 93% accuracy, even with molecules not involved in the building of the path sets, which supports the assertion that these are generally applicable sets.

Further validation of the worth of these path sets can be undertaken by applying them to activity prediction using diverse molecular datasets. A number of steps in the study, of necessity, used an intuitive heuristic approach. There remains potential for further investigation into, and formalizing of, these techniques. The path lengths used were limited by the computing resources available. It was observed that the longer paths were strong contributors to the predictive power of the descriptors. Since the path lengths were limited to 12 there is scope to extend this study, with sufficient computing power, to identify the optimum range of path lengths for prediction. In this study the descriptors only registered the existence or absence of a path in a molecule. There

is potential, in further studies, to capture more information by storing the frequency of occurrences of each path in a molecule and applying regression algorithms.

10 Acknowledgment

The authors would like to express our thanks to Ms Ruchi Mittal of the Sansom Institute at the University of South Australia for the provision of the LitSet data.

References

- Bezem G. J & van Leeuwen J. (1987) 'Enumeration in Graphs' *Technical Report RUU-CS-87-7*, *Rijksuniversiteit Utrecht* from <http://www.cs.uu.nl/research/techreps/repo/CS-1987/1987-07.pdf> accessed May 2009.
- Breiman, Leo. (2001) 'Random Forests' *Machine Learning*, **45**(1), pp. 5–32.
- Cheng, Jie. (2001) 'KDD CUP 2001 Task 1: Thrombin' <http://www.sigkdd.org/kddcup/site/2001/files/Cheng.pdf>.
- Gordon H Danielson. (1968) 'On Finding the Simple Paths and Circuits in a Graph' *IEEE Transactions on Circuit Theory*, **1968**(Sept), p. 294.
- Drummond C. & Holte R. C. (2003) 'C4.5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling' *Workshop on Learning from Imbalanced Datasets II*, **ICML** (2003) .
- Fawcett Tom (2007) 'An introduction to ROC analysis' *Pattern Recognition Letters*, **27** (8) pp. 861–874.
- Frank Eibe, Hall Mark, Trigg Len, Holmes Geoffrey & Witten, Ian H (2004) 'Datamining in Bioinformatics using Weka' *Bioinformatics*, **20** (15) pp. 2479–2481.
- César Ferri, Peter Flach & José Hernández-Orallo (2002) 'Learning Decision Trees Using the Area Under the ROC Curve' *Proceedings of the Nineteenth International Conference on Machine Learning*, Morgan Kaufmann Publishers inc. pp. 139–146.
- von Grotthuss M, Koczyk G, Pas J, Wyrwicz LS & Rychlewski L (2004) 'Ligand.Info Small-Molecule Meta-Database' *Combinatorial Chemistry & High Throughput Screening*, **7** (8) pp. 757–761.
- Hanser Th, Jauffret Ph & Kaufmann G. (1996) 'A New Algorithm for Exhaustive Ring Perception in a Molecular Graph' *Journal of Chemical Information and Computer Sciences*, pp. 111–135.
- Huang J. & Ling C. (2005) 'Using AUC and accuracy in evaluating learning algorithms' *Knowledge and Data Engineering, IEEE Transactions on*, **17** (3) pp. 299–310.
- Jovanoski, Viktor & Lavrač, Nada (2001) 'Classification Rule Learning with APRIORI-C' *Progress in Artificial Intelligence*, **36** (6) pp. 1146–1152.
- Kohavi R. & Quinlan R (2002) 'Decision Tree Discovery' *Handbook of Data Mining and Knowledge Discovery*, Will Klossen and Jan M. Zytkow eds. **Ch 16** pp. 267–276.
- Kubinyi, Hugo. (1997) 'QSAR and 3D QSAR in drug design Part 1: methodology.' *Drug Discovery Today*, **2**(11), pp. 457–467.
- Kubinyi, Hugo. (1998) 'Comparative Molecular Field Analysis (CoMFA)' *The Encyclopedia of Computational Chemistry*, Schleyer, P. v. R., Allinger, N. L., Clark, T., Gasteiger, J., Kollman, P. A., Schaefer III, H. F., & Schreiner, P. R., Eds., **Vol 1** pp. 448–460.
- Mittal Ruchi, McKinnon Ross & Sorich Michael (2008) 'Effect of steric molecular field settings on CoMFA predictivity' *Journal of Molecular Modeling*, **14** (1) pp. 59–67.
- NCI Aids Antiviral Screen (2004) http://dtp.nci.nih.gov/docs/aids/aids_data.html
- Nehmé Kamal, Petko Valtchev, Mohammed H. Rouane & Robert Godin (2005) 'On Computing the Minimal Generator Family for Concept Lattices and Icebergs' *ICFCA 2005, LNCS 3403*, B. Ganter and R Godin (Eds.) **2005** (48) pp. 192–207.
- Rush T. S. 3rd, Grant J. A, Mosyak L. & Nicholls A. (2005) 'A Shape-Based 3-D Scaffold Hopping Method and Its Application to a Bacterial Protein-Protein Interaction' *Journal of Medicinal Chemistry*, **2005** (48) pp. 1489–1495.
- Schnecke V. & Boström (2006) 'Computational chemistry-driven decision making in lead generation' *Drug Discovery Today*, **11** (1-2) pp. 43–50.
- Jun Xu & James Stevenson (2000) 'Drug-like Index: A New Approach To Measure Drug-like Compounds and Their Diversity' *Journal of Chemical Information and Computer Sciences*, **40** (5) pp. 1177–1187.
- Ling Xue & Jurgen Bajorath (2000) 'Molecular Descriptors in Chemoinformatics, Computational Combinatorial Chemistry and Virtual Screening.' *Combinatorial Chemistry & High Throughput Screening*, **2000**(3), pp. 363–372.

Single Document Semantic Spaces

Jorge Villalon

Rafael A. Calvo

School of Electrical & Information Engineering
The University of Sydney
Email: {villalon,rafa}@ee.usyd.edu.au
<http://www.weg.ee.usyd.edu.au>

Abstract

Latent Semantic Analysis (LSA) has been successfully used in a number of information retrieval, document visualization and summarization applications. LSA semantic spaces are normally created from large corpora that reflect an assumed background knowledge. However the right size and coverage of the background knowledge for each application are still open research questions. Moreover, LSA's computational cost is directly related to the size of the corpus, making the technique inviable in many cases. This paper introduces a technique for creating semantic spaces using a single document and no background knowledge, which cuts computational cost and is domain independent. Single document semantic spaces' reliability was evaluated on a collection of student essays. Several semantic spaces generated from large corpora and single documents were used to compare how essays are represented. The distance between consecutive sentences in the essays changes between semantic spaces, but the rank of the distances is preserved. The results show that high correlations (0.7) of ranked distances between sentences can be achieved on the different spaces for the weight schemes evaluated. This has important implications for the applications discussed.

Keywords: Single Document Semantic Space, Latent Semantic Analysis, LSA, background knowledge, corpus size

1 Introduction

A common problem encountered in information retrieval, document analysis and visualization applications is that people use words for their collective meaning and not just for the literal term. Linguistically the difficulties introduced are explained by the synonymy and polysemy problems. The former refers to the many ways of expressing the same concept, where people adapt their vocabulary based on the topic being discussed, or on the particular background knowledge (both of the writer and/or the reader). The latter refers to the many meanings that a word can have, meanings that humans are able to disambiguate using information about the topic being discussed or other contextual information. Synonymy and polysemy are known to affect the accuracy of computer systems that use terms (instead of concepts) as the main way of representing information.

Copyright ©2009, Australian Computer Society, Inc. This paper appeared at the Eighth Australasian Data Mining Conference (AusDM 2009), Melbourne, Australia. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 101, Paul J. Kennedy, Kok-Leong Ong and Peter Christen, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

In information retrieval in particular synonymy affects recall and polysemy affects precision.

Latent Semantic Analysis (LSA) is a statistical dimensionality reduction technique proposed by Deerwester et al. (Deerwester et al. 1990) to address these issues by indexing documents based on 'concepts' rather than terms. This requires a semantic representation for the corpus of documents and queries. LSA starts with a term-document matrix and uses Singular Value Decomposition (SVD) to create a *semantic space* where the distances between terms and/or documents reflect a 'semantic' proximity. When LSA is used for information retrieval tasks user queries are *projected* in the semantic space as *pseudo documents*.

The 'concepts' in this semantic space are features of the set of documents used to create the space. For example, the collection of books that kids in primary school are likely to have read can be used to create the semantic space that best fits the way in which primary school kids communicate. This 'shared' knowledge representation is useful in information retrieval and learning technologies. For example, LSA has been used to analyze the quality of students' essays, using papers and books that the student should have read as background knowledge and the essays as queries. In fact, systems such as e-Rater and Intelligent Essay Assessor that use LSA for automatic essay grading (Miller 2003, Burstein et al. 2003) use semantic spaces to mark essays by calculating the distances between previously marked essays and the one to be marked. These automatic grading tools have shown to be very reliable (Shermis & Burstein 2003) and are commercially used in standardized tests.

LSA has also been used to measure deeper quality patterns in essays, such as discourse coherence (Foltz 2007). By measuring the distances between consecutive sentences and/or paragraphs, possible breaks in coherence are identified. These measures have been found to correlate positively with the quality of essays (Graesser et al. 2004).

In the above applications the concepts in a single essay are described as a function of the shared concepts in the semantic space created from the background knowledge. However, the right size and coverage of the background knowledge for each application are still open research questions. Moreover, LSA's computational cost is directly related to the size of the corpus, making it many times inaccessible for the final user.

Other dimensionality reduction techniques that do not make any assumptions regarding the background knowledge are more appropriate in some applications. For example, Gong and Liu (Gong & Liu 2001) proposed a generic text summarization technique that aims to summarize a document by selecting sentences that are important, and yet different from each other. They used a variation on LSA that creates a semantic space based on the single document to be summa-

rized. This approach was shown to be useful on sets of documents that cover a broad set of topics (i.e. news stories). Other applications that would benefit from this approach are automatic concept mapping on essays (Villalon & Calvo 2008), and clustering search results (Osinski 2006), both covering broad sets of topics. The distances between documents or sentences produced by these two types of LSA spaces will be different, but this paper studies what the approaches have in common and shows how the ranked lists of consecutive sentences mapped by the different approaches correlate.

This paper contributes an analysis of semantic spaces generated with a single document, and how they compare with those generated from small and large corpora. The results provide evidence that the different approaches produce similar ranked list of distances between consecutive sentences, however single document semantic spaces require much less computational power.

Section 2 presents LSA and previous work analyzing the effect of small corpora including single document semantic spaces. Section 3 describes the implementation issues in single document semantic spaces, while Section 4 describes the methodology and results of the evaluation. Our evaluation used real-world essay corpora and compared how different spaces produce different ranked distances between consecutive sentences. Section 5 presents a discussion of its implications.

2 Previous work

This section briefly describes the mathematical background of LSA, and discusses the two areas of previous work that are relevant for this study: Studies analyzing the effect of the corpus size, particularly very small corpora, and the use of LSA in a single document to obtain meaningful patterns.

2.1 Latent Semantic Analysis

LSA defines the semantic space of a term-by-document (or term-by-sentence) matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$ (that can be called the knowledge base) by decomposing it using Singular Value Decomposition as:

$$\mathbf{X}_k = \mathbf{U}_k \Sigma_k \mathbf{V}_k^T$$

where $\mathbf{U}_k \in \mathbb{R}^{n \times k}$, $\Sigma_k \in \mathbb{R}^{k \times k}$, $\mathbf{V}_k \in \mathbb{R}^{m \times k}$ and $k < \min(m, n)$

In this representation the m columns X_i represent the weighted term-frequency vectors (of size n) of each of the documents used to create the *semantic space*. The column vectors of orthonormal matrices \mathbf{U}_k and \mathbf{V}_k are the left and right singular vectors respectively. Σ_k the non-negative diagonal matrix of the k biggest singular values sorted in descending order. The rows of \mathbf{U}_k and \mathbf{V}_k can be interpreted as the coordinates of points that represent terms and documents respectively in the k dimensional space.

If new documents need to be represented in this semantic space, they can be represented as $\mathbf{d} \in \mathbb{R}^n$ and projected on the k -dimensional space as:

$$\hat{\mathbf{d}} = \mathbf{d}^T \mathbf{U}_k \Sigma_k^{-1}$$

The result $\hat{\mathbf{d}}$ is a k -dimensional vector that can be compared with other documents in the original knowledge base corpora or with other query documents. The criteria to decide the value of k still remains an open question for LSA and is usually set for individual experiments (Dumais 1991, Landauer et al. 1998, Haley et al. 2005).

2.2 Use of small corpora

Most studies on LSA have used large corpora to create semantic spaces, however the right size for a corpus remains an open question. In a recent report by Giesbers et al. (Giesbers et al. 2006) they argued that it is not clear what a *small* or *large* corpus is, the same applies for minimum or maximum requirements. In LSA's seminal paper, Deerwester suggested that a *reasonable size* for a corpus should be between 1,000-2,000 documents and 5,000-7,000 terms (Deerwester et al. 1990), this is not surprising because LSA's original purpose was to improve IR accuracy by finding the underlying concept in the words of a user query. The technique's accuracy is based on redundancy in the corpora, the premise being that terms that appear together are conceptually related.

Gong and Liu (Gong & Liu 2001) serendipitously used a single document to generate a semantic space for automatic summarization but did not compare it to other techniques or analyze the trade-offs involved. The technique has been used to create visualizations (Stephen O'Rourke 2009), labeled clusters (Osinski 2006), and automatic feedback for students (Villalon et al. 2008) but again the technique itself was not analyzed in detail and questions remain on how it works.

In the summarization work, semantic spaces created from a single document were used to extract the most relevant sentences in a document identifying topics in the document and selecting the most representative sentences for each topic. Semantic spaces were created using each of the document's paragraphs or sentences, with each singular vector obtained from the SVD representing a different topic (Steinberger et al. 2007).

2.3 Measuring coherence with LSA

One important aspect of the quality of essays is coherence (or cohesion), which reflects how the author links related pieces of information to create the essay's structure. Foltz explains that for a text to be coherent it requires a high quality "overlap and transitions of the meaning as it flows across the discourse", and LSA is able to model this phenomenon "by measuring the semantic similarity of one section of text to the next" (Foltz 2007).

The measurement of coherence using LSA was first proposed by Foltz et al. (?), in their study they calculated the essay's coherence as the average distance between consecutive sentences. They found that the LSA measure correlated better with human comprehension of the text than other automatic measures for coherence such as word overlap and readability indices. Another study by Higgins et al. (?) used an LSA like algorithm to calculate coherence as a quality measure for automatic essay grading. They calculated the semantic distance between sentences, between discourse segments (usually paragraphs) and the prompt (text of the essay question), and between each discourse segment and the essay thesis (one of the discourse segments). They found that all three aspects correlated higher than a random baseline. Another study by Graesser et al. used LSA to calculate several distances between sentences, paragraphs and the whole document, including consecutive sentences and consecutive paragraphs to measure coherence and provide feedback to students (Graesser et al. 2004). These measures were used to reliably identify the writing style of different authors (?). Many other examples of measuring coherence with LSA can be found in (Foltz 2007).

Term Weight	Formula
Tf	tf_{ij}
TfIdf	$tf_{ij} * \log_2(\frac{ndocs}{df_i}) + 1$
LgEnt	$\log(1 + tf_{ij}) * (1 - \sum_j \frac{p_{ij} * \log(p_{ij})}{\log(ndocs)})$ where $p_{ij} = \frac{tf_{ij}}{gf_i}$

Table 1: Term weights schemes and their formulas

3 Implementation of Single Document Semantic Spaces

This section describes the technical details on the construction of the single document semantic spaces. The technique requires several parameters in each step: document pre-processing, term weighting and dimensionality reduction.

3.1 Document pre-processing

The pre-processing step corresponds to the extraction of the terms from the document. First, all documents were split into paragraphs using a simple matching to the newline character. Second, each paragraph was split into sentences using Sun's standard Java text library. Third, each sentence, paragraph and the whole document were indexed using Apache's Lucene search engine, which performed the tokenization, stemming, and stop words removal¹. The result of this step was a Lucene index storing the term frequency vectors for each passage (full document, paragraph and sentence).

3.2 Term weighting

Term frequency (TF) and document frequency (DF) were used to weight terms and discard those that only appeared once or in just one passage. Once all unwanted terms were discarded, the term-passage matrix A was formed, each of its values a_{ij} indicates the frequency of the i_{th} term for the j_{th} text passage. However, most LSA applications use more sophisticated alternatives to the simple term frequency measure, which corresponds to a combination of a local and global measure for each term appearance and are known as term weight schemes.

Table 1 shows the local and global weight schemes used in this study with their corresponding formulas, where tf_{ij} is the number of times the term i appears in the document j , df_i is the document frequency of term i , which is the number of documents where the term appears, and gf_i is the global frequency, which is the total number of times the term appears in the whole collection.

3.3 Dimensionality reduction

Reducing the dimensions of the space is a key step in LSA. It is in this process where the semantic relationships between terms and passages are surfaced. Previous LSA studies have suggested a value for k between 200 and 600, however this rule only applies to large corpora and it would not work for essays, which are generally shorter than 200 paragraphs or sentences. As the value of k has to be lower than the minimum between passages and different terms,

the limits have to be set as a function of the essay content.

Other factor analysis techniques that perform dimensionality reduction, such as Principal Component Analysis, have defined criteria to find k . These criteria can be classified in three: Ad-hoc but intuitively plausible, and statistically based, both with and without distributional assumptions (?). The first one includes selecting k based on the singular values in Σ , like a percentage of the cumulative variance, or all values above 1 (Kaiser's rule). The second includes all singular values until the data fits an assumed distribution. The third one also includes singular values until a criterion such as cross-validation or bootstrapping is fulfilled.

However, deciding an appropriate value of k for single document semantic spaces presents added complexities. In LSA there are no theoretical frameworks that explain the distribution of singular values in text corpora, least of all in single document spaces. Moreover, short essays usually have a lower number of sentences than different terms, and each one has a different length. Therefore k can only take values between 1 (only one dimension) and the total number of sentences in the essay (no dimensionality reduction).

In order to study how dimensionality reduction in single document spaces affects its passage distances, k will be evaluated from its minimum to its maximum values. Therefore k was defined as a percentage of the maximum number of dimensions, which corresponds to the minimum between the number of passages and the number of words, varying from 5% to 100% in steps of 5% each.

4 Evaluation

Essays (N=43) collected as a diagnostic activity for first year university students were used to evaluate different semantic spaces. The essays were written in an activity where students first read three short papers on the topic of *English as a global language*, and then answered two questions: Is English becoming a global language? Is this a positive development? The three readings were 874, 812 and 888 words long respectively (858 average).

Distances between consecutive sentences were calculated for each essay using different semantic spaces created from different background knowledge: Single document, large corpora and prescribed readings for the activity. Using the notation defined earlier, each of these three semantic spaces will produce different U_k and Σ_k^{-1} .

Each sentence in an essay (\mathbf{d}) was projected on the particular semantic space being studied, producing a $\hat{\mathbf{d}}$ vector². Having the coordinates of each sentence on a semantic space, the distance between each consecutive $\hat{\mathbf{d}}$ was then calculated using a cosine function.

The distances change for LSA spaces generated with different background knowledge, but for the applications of concern in this study only the relationships (relative distances) between the sentences are required. The distances were then ranked, and the ranking correlation was calculated using Spearman's ρ . The statistical significance was calculated using a permutation-test (also known as randomization-test and exact test) (Zar 1972). All statistics were calculated using the JSC Java library for statistical computation³. The correlation for the collection of essays was calculated as the average of the correlations for each essay.

¹The Snowball analyzer was used, which in turn uses the Porter's stemmer.

²For the single document space there's no need to project the sentences because they actually formed the space.

³<http://www.jsc.nildram.co.uk/>

Corpus	06th Grade	09th Grade	12th Grade	College	Psych
03rd Gr	0.93	0.9	0.88	0.85	0.79
06th Gr		0.97	0.95	0.92	0.8
09th Gr			0.97	0.95	0.8
12th Gr				0.98	0.81
College					0.81

Table 2: Spearman's rank correlation for each LSA colorado corpora

Spearman's correlation reliability is affected by ties groups (two or more distances with the same value), because these distances are interchangeable in the ranking without affecting the correlation (Zar 1972). The bigger the group of ties, the bigger the effect on the reliability, also more than one group of ties can occur in each document, hence the average size of ties groups for each document was also analyzed using the same conditions as for the distances (when no ties were found in a document, its average ties group size was assigned to 1).

4.1 Large corpora

The distances between consecutive sentences of each essay were calculated with LSA spaces produced by six different corpora in the Colorado LSA website (Dennis 2007). The corpora used included: five sets produced by TASA (Touchstone Applied Science Associates, Inc.) 3rd Grade (6,974 documents and 29,315 terms), 6th Grade (17,949 documents and 55,105 terms), 9th Grade (22,211 documents and 63,582 terms), 12th Grade (28,882 documents and 76,132 terms) and 1st Year College (37,651 documents and 92,409 terms); and 'Psych' (13,902 documents and 30,119 terms). For each corpora the first 300 dimensions (k) were preserved.

Table 2 shows the average rank correlation of the 43 essays projected on the different TASA (Touchstone Applied Science Associates, Inc.) corpus and 'Psych'. Since the reading materials that students in 6th year are supposed to know include those of 3rd year, every TASA corpus includes those of lower literacy levels, and this can be seen in how the bigger the difference in content included the smaller the correlation.

The average ties group size produced by the TASA and 'Psych' corpora was 1.968, which means that in average no more than two consecutive sentences got the same distance, making the rank correlations reliable.

4.2 Single document semantic spaces

The essays used in this study were written by 1st year students, so the TASA 1st year college corpus was used to compare the ranked distances. The essays are between 9 and 33 sentences long, with an average of 21 and median 22. As k was defined as a variable number of dimensions for each document (from 5% to 100% in 5% steps), therefore k 's real value ranged roughly between 1 and 20, but varying from essay to essay.

The first analysis was the average distance between consecutive sentences, calculated for all term weighting schemes and k values. Figure ?? shows that with $k = 5\%$ of the dimensions, there was only one

dimension in which to project the sentences, therefore most distances were 1 (cosine of 0), meaning all sentences were almost identical. Distances dropped rapidly from $k = 5\%$ as more dimensions were included, slowing down around $k = 35\%$ of the dimensions. With $k = 100\%$, there was no dimensionality reduction, therefore no semantic relationships were discovered. If two sentences had no common terms, they were linearly independent and their distance was 0 (cosine of $\frac{\pi}{2}$).

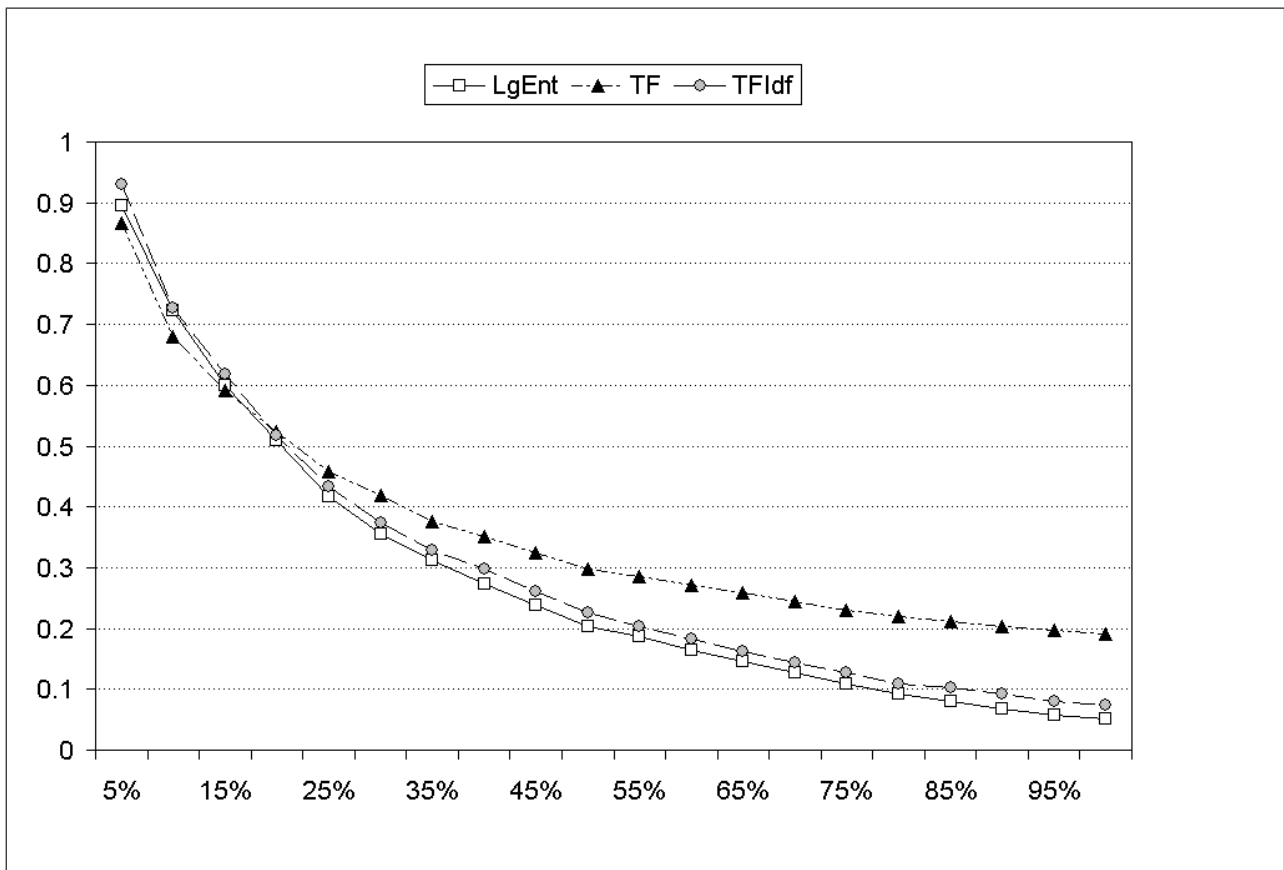
The average distance between consecutive sentences dropped faster for both LogEntropy and TFIdf. Global weights (Entropy and Idf) are used to smooth the effect of rare terms and/or too long documents in large corpora, however in single documents semantic spaces most terms have low frequencies (therefore there are no rare terms), and the length of sentences is limited (therefore there are no long documents). Therefore using them produced an information loss that made the sentences more dissimilar.

The second analysis was the average ties group size calculated for all k values and weighting schemes. Figure ?? shows that with $k = 5\%$, an average group size of 9 ties was produced, which corresponds to almost 50% of the sentences. As more dimensions were added the groups of ties diminish, almost disappearing between $k = 20\%$ and $k = 50\%$, and starting to grow again reaching an average of 4 when all dimensions were included. This phenomenon was caused by the distances between sentences, because ties are produced by too similar or too dissimilar sentences.

The final analysis was the average Spearman's correlation ρ of ranked distances for the 43 essays projected both on the single document and the TASA 1st year college semantic spaces, calculated for all k values and term weighting schemes. The correlation divided by the average ties group size was also calculated (ρ' in the graph), to illustrate the loss of reliability in the calculation of ρ . Figure 1 shows that the TF weighting scheme is the best, with values above 0.6 when $k = 40\%$, and continued growing as dimensions were added. The adjusted ρ' started falling around $k = 55\%$ due to the appearance of ties, however the average ties group size remained below 2 (the average ties group size for the large corpora) until $k = 80\%$, when ρ reached 0.7. Statistical significance was also calculated but not shown in the graph. For both LogEntropy and TFIdf it was never below 0.05 (95% confidence), until almost all dimensions were included ($k = 90\%$), results that due to the ties produced were not reliable to make a fair comparison. TF on the other hand rapidly achieved a significance below 0.05 and it was stable between $k = 35\%$ and $k = 75\%$.

4.3 Prescribed readings

The distances between consecutive sentences were also calculated using a semantic space created from a small corpus containing the prescribed readings (125 sentences and 199 terms). The single document space correlated with the readings space almost identically than with the TASA corpus (0.71, $p < 0.001$) even though the readings space was three orders of magnitude smaller than the TASA corpus. This could be explained by the specificity of the readings, because the essays were expected to have similar content to them. Finally the correlation between the readings space and the TASA corpus was slightly higher (0.76, $p < 0.001$) than the single document corpus. This could be explained by associations between terms that were explicit both in the readings and the TASA corpus, but implicit in the students' essay.

Figure 1: Average sentence distance vs k .

5 Conclusion

This paper questioned the premise that a knowledge base is needed to generate semantic spaces used in summarization, visualization and other applications. Different approaches to constructing semantic spaces were evaluated.

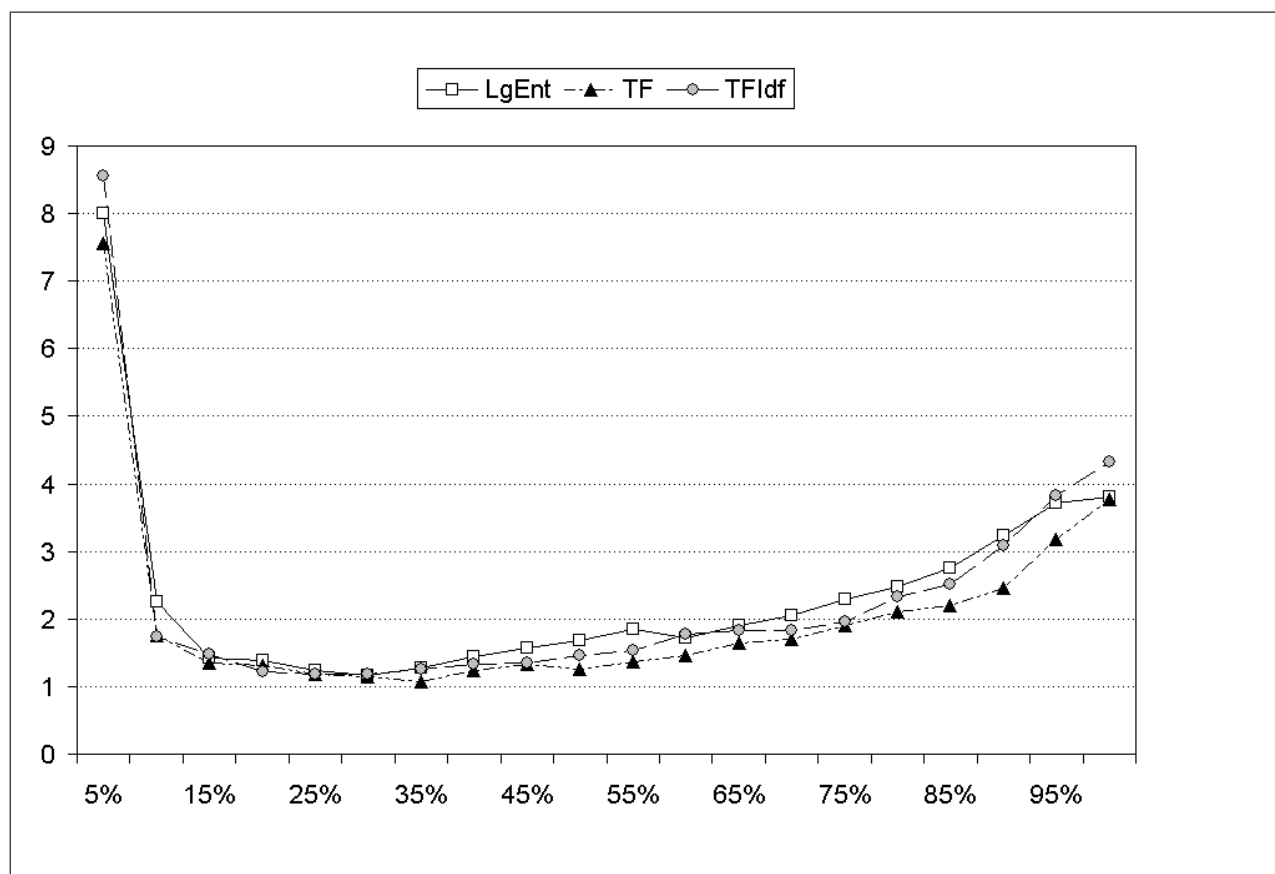
First we showed that the distances between consecutive sentences change significantly between the representations produced by different background knowledge. This makes certain applications unfeasible or more demanding because customized knowledge bases are required.

The results show that the collection used to create the semantic spaces must be taken into account for each application. For example, the differences between using a collection of documents such as those read by a 3rd year primary school student and those read by college student are not substantial ($\rho = 0.85$) while they are significant when using a collection of medical papers (a topic not related to the topic of the essays).

We also evaluated how the inter-sentence distance on a semantic space generated only using the sentences of the essay correlates to those generated with the College database. We found that good rank correlation can be achieved (up to 0.7) using a variable number of dimensions, 75% of the maximum possible.

References

- Burstein, J., Marcu, D. & Knight, K. (2003), 'Finding the write stuff: automatic identification of discourse structure in student essays', *IEEE Intelligent Systems* **18**, 32–39.
- Deerwester, S., Dumais, S., Furnas, G., Landauer, T. K. & Harshman, R. (1990), 'Indexing by latent semantic analysis', *Journal of the American Society for Information Science* **41**(6), 391–407.
- Dennis, S. (2007), How to Use the LSA Web Site, in T. K. Landauer, D. S. McNamara, S. Dennis & W. Kintsch, eds, 'Handbook of Latent Semantic Analysis', Routledge, pp. 71–85.
- Dumais, S. (1991), 'Improving the retrieval of information from external sources', *Behavior Research Methods, Instruments, & Computers* **23**, 229–236.
- Foltz, P. W. (2007), Discourse coherence and lsa, in T. K. Landauer, D. S. McNamara, S. Dennis & W. Kintsch, eds, 'Handbook of Latent Semantic Analysis', Routledge, pp. 167–185.
- Giesbers, B., Rusman, E. & van Bruggen, J. (2006), State of the art report in knowledge sharing, recommendation and latent semantic analysis, Technical report, Cooper Consortium.
- Gong, Y. & Liu, X. (2001), Generic text summarization using relevance measure and latent semantic analysis, in 'Proc. of the Int. Conf. on Research and development in information retrieval', ACM, pp. 19–25.
- Graesser, A. C., McNamara, D. S., Louwerse, M. M. & Cai, Z. (2004), 'Coh-metrix: Analysis of text on cohesion and language', *Behavior Research Methods, Instruments, & Computers* **36**, 192–202.
- Haley, D. T., Thomas, P., Roeck, A. D. & Petre, M. (2005), A research taxonomy for latent semantic analysis-based educational applications, Technical report, Department of Computing, Faculty of Mathematics and Computing, The Open University.

Figure 2: Average ties groups size vs k

Landauer, T. K., Foltz, P. W. & Laham, D. (1998), 'An introduction to latent semantic analysis', *Discourse processes* **25**(2 & 3), 259–284.

Miller, T. (2003), 'Essay assessment with latent semantic analysis', *Journal of Educational Computing Research* **28**(3).

Osinski, S. (2006), Improving quality of search results clustering with approximate matrix factorisations, in 'Proceedings of the 28th European Conference on Information Retrieval Research (ECIR)', pp. 167–178.

Shermis, M. D. & Burstein, J. (2003), *Automated essay scoring*, Lawrence Erlbaum Associates.

Steinberger, J., Poesio, M., Kabadjov, M. A. & Jezek, K. (2007), 'Two uses of anaphora resolution in summarization', *Information Processing and Management* **43**, 1663–1680.

Stephen O'Rourke, R. A. C. (2009), Semantic visualisations for academic writing support, in '14th Conference on Artificial Intelligence in Education', IOS Press.

Villalon, J. & Calvo, R. (2008), Concept Map Mining: A Definition and a Framework for Its Evaluation, in 'IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT'08', Vol. 3.

Villalon, J., Kearney, P., Calvo, R. & Reimman, P. (2008), Glosser: Enhanced feedback for student writing, in 'Proceedings of the International Conference on Advanced Learning Technologies'.

Zar, J. H. (1972), 'Significance testing of the spearman rank correlation coefficient', *Journal of the American Statistical Association* **67**, 578–580.

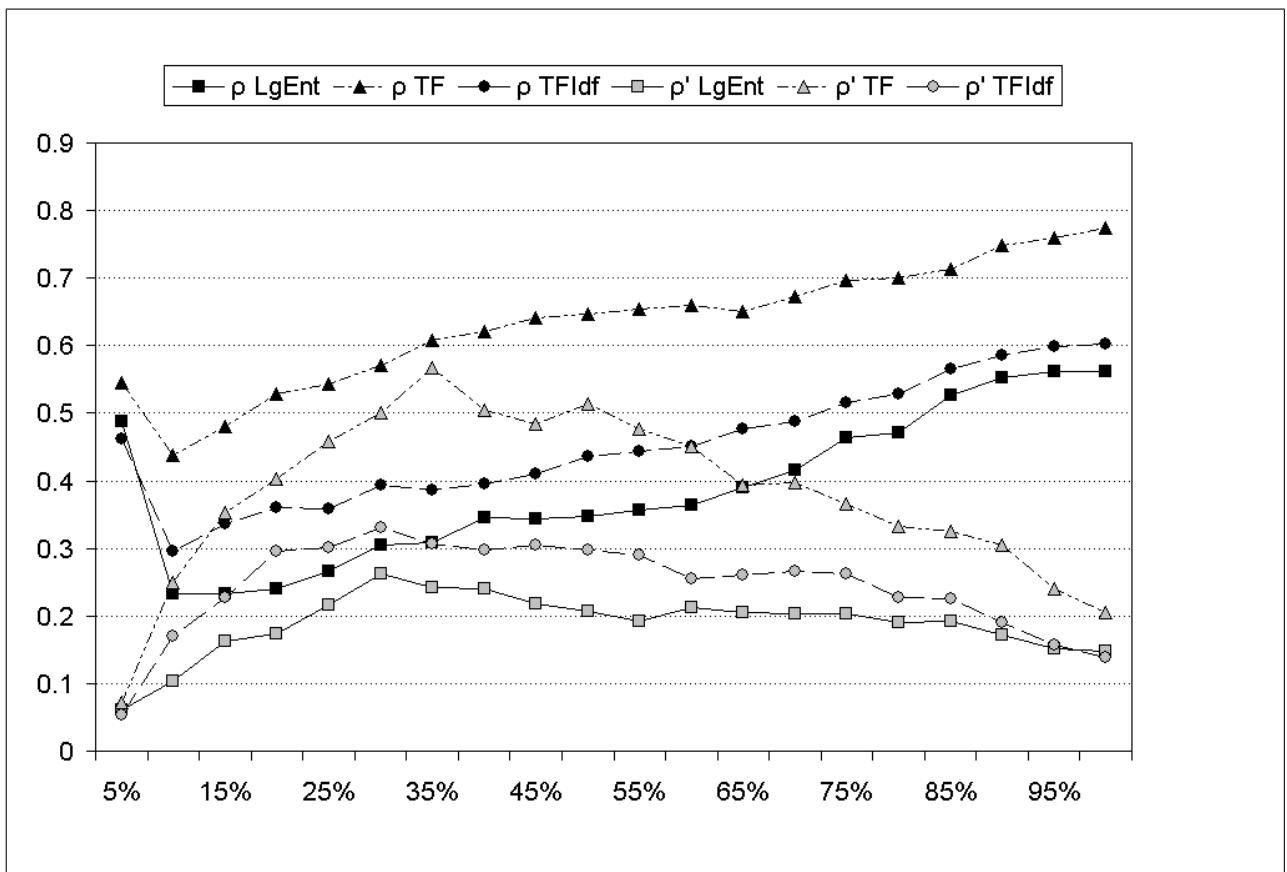


Figure 3: Average Spearman ρ and ρ' for the 43 essays projected on both the single document and TASA 1st year college semantic spaces vs k .

Efficient Mining of Top-k Breaker Emerging Subgraph Patterns from Graph Datasets

Min Gan

Honghua Dai

School of Information Technology
Deakin University
Melbourne, Victoria 3125, Australia
Email: {min.gan.au, honghuadai}@gmail.com

Abstract

This paper introduces a new type of discriminative subgraph pattern called breaker emerging subgraph pattern by introducing three constraints and two new concepts: base and breaker. A breaker emerging subgraph pattern consists of three subpatterns: a constrained emerging subgraph pattern, a set of bases and a set of breakers. An efficient approach is proposed for the discovery of top-k breaker emerging subgraph patterns from graph datasets. Experimental results show that the approach is capable of efficiently discovering top-k breaker emerging subgraph patterns from given datasets, is more efficient than two previous methods for mining discriminative subgraph patterns. The discovered top-k breaker emerging subgraph patterns are more informative, more discriminative, more accurate and more compact than the minimal distinguishing subgraph patterns. The top-k breaker emerging patterns are more useful for substructure analysis, such as molecular fragment analysis.

Keywords: Breaker emerging subgraph patterns, discriminative patterns, graph mining.

1 Introduction

As an abstract data structure, graphs are suitable for representing any objects and their relationships. A graph is a set of vertices and edges, where a vertex represents an object, and an edge between two vertices represent that a relationship exists between the two vertices. In real world, there exist large amounts of data that can be represented as graphs, such as molecular structures, Web-link structures, biological networks, transport networks and social networks. Graph mining mainly studies how to discover knowledge from graph data. In recent years graph mining has become an active research field. Many graph mining methods have been proposed for discovering various patterns from graph data. No matter what methods are used and what patterns are discovered, many graph mining tasks need to conduct a key operation, graph comparison, which detects two kinds of information: graph similarity and graph dissimilarity. This paper focuses on graph dissimilarity.

Graph dissimilarity reflects the difference between two graphs or two classes of graphs. Conventional graph matching metrics such as graph edit distance (Sanfeliu et al. 1983), maximal common subgraphs

(McGregor 1982) and subgraph isomorphism can be used to measure the dissimilarity (as well as the similarity) between two graphs. However, these metrics are not applicable to measuring the dissimilarity between two contrasting classes of graphs, which is a key issue in graph mining. From an application point of view, in graph mining, there exist many cases in which one needs to detect the dissimilarity between two contrasting classes of graphs. For example, in drug analysis, medical experts detect molecular differences between two classes of drug components (strong side effect vs. weak side effect) to explore the molecular mechanism of the side effect. In e-commerce website analysis one detects differences of Web access behaviors between two classes of visitors (purchaser vs. non-purchasers or males vs. females) to improve website organization or provide customized Web-link structures. From the point of view of data mining theory, the differential information between two contrasting classes of data is crucial for many mining tasks such as classification. In order to distinguish from the dissimilarity between two individual graphs, in this paper “graph class dissimilarity” is used to denote the dissimilarity between two classes of graphs.

Graph class dissimilarity is usually represented as discriminative subgraph patterns. Therefore, the first problem in detecting graph class dissimilarity is: which patterns are the best to be used for identifying graph class dissimilarity. The second problem is how to discover the patterns efficiently. Discriminative subgraph patterns can be classified into two categories: one is discriminative individual (connected) subgraph patterns, and the other is discriminative multiple subgraph patterns (a pattern consists of one or multiple connected subgraphs). The former is normally used for individual substructure analysis. The latter is usually more discriminative than the former and is used for effective classification. The two types of patterns are more complementary than competitive. In this paper, we focus on discriminative individual subgraph patterns.

Most existing discriminative patterns are only for simple types of data, such as transactional data and relational data, and few discriminative patterns for graph data have been proposed. As an important discriminative pattern, emerging pattern (EP) (Dong et al. 1999) has been proved to be of strong discriminating power for distinguishing between two classes of data, and has broad applications, such as construction of accurate classifiers (Ramamohanarao et al. 2006). In recent years, researchers have extended the discovery of emerging patterns from simple types of data to graph data, and proposed two kinds of patterns: contrast subgraph pattern (CSP) (Ting et al. 2006) and distinguishing subgraph pattern (DSP) (Zeng et al. 2008). Recently Fan et al. proposed a general discriminative pattern, discriminative and essential frequent pattern (DEFP) (Fan et al. 2008),

which applies to various types of data including graph data.

However, we found that none of CSP and DSP include the most discriminative individual subgraph patterns exactly, and both of them have some drawbacks as analysed in the next section. The DEFP is essentially discriminative and has been applied to effective classification (Cheng et al. 2008), but as a kind of discriminative multiple subgraph pattern, it is not applicable to individual substructure analysis. Our study aims at introducing a more accurate and more informative discriminative individual subgraph pattern, and devising an efficient mining algorithm. In this paper, we introduce a new type of discriminative subgraph pattern called breaker emerging subgraph pattern (BESP), and devise an efficient algorithm to discover the top-k BESP from graph datasets.

The rest of this paper is organised as follows. Related work is reviewed and analysed in Section 2. Motivations are illustrated in Section 3. Section 4 defines the breaker emerging subgraph pattern. Section 5 proposes an efficient algorithm for mining top-k BE-SPs. Experimental results are presented in Section 6. Conclusions and future work are included in Section 7.

2 Related Work

In this section, we provide a brief summary of the related work on emerging patterns, contrast subgraph patterns and distinguishing subgraph patterns.

2.1 Emerging Pattern

The emerging pattern (EP) was originally proposed by Dong et al. (1999). An EP is defined as an item-set X whose support increases significantly from one dataset D_N to another, D_P , where the increasing degree of the support is measured by growth rate, which is defined as

$$GR_{D_N \rightarrow D_P}(X) = \begin{cases} 0 & \text{if } \sup_N(X) = 0 \\ & \text{and } \sup_P(X) = 0 \\ \infty & \text{if } \sup_N(X) = 0 \\ & \text{and } \sup_P(X) \neq 0 \\ \frac{\sup_P(X)}{\sup_N(X)} & \text{otherwise} \end{cases} \quad (1)$$

where $sup_P(X)$ is the support of itemset X in \bar{D}_P , which equals $count_P(X)/|D_P|$, $count_P(X)$ is the total number of transactions in D_P that contain X , and $|D_P|$ is the total number of transactions in D_P . Similarly $sup_N(X)$ represents the support of X in D_N , which equals the number of transactions in D_N that contains X over the total number of transactions in D_N , denoted by $|D_N|$ (Dong et al. 1999).

Given the minimal growth rate threshold Min_GR , EPs from D_N to D_P are itemsets whose growth rates are no less than Min_GR (Dong et al. 1999). The higher GR of an EP is, the more discriminative and more significant the pattern is. In this paper the growth rate is also adopted to evaluate the discriminating power of a pattern.

Although the EP was originally defined on items by Dong et al. (1999), it applies to any other types of data including graph data. When it is defined on graph data, the EP can be called emerging subgraph pattern (ESP).

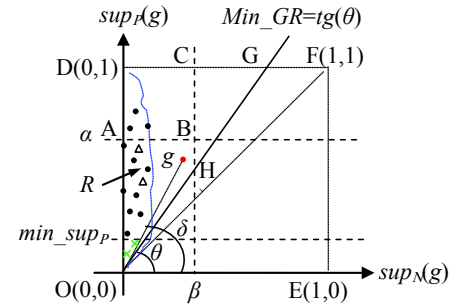


Figure 1: The support plane of emerging subgraph patterns

2.2 Contrast Subgraph Pattern

Contrast subgraph patterns (CSPs) (Ting et al. 2006) are defined as subgraphs¹ that appear in one class of graphs D_P , but never appear in another class of graphs D_N . A CSP is minimal if none of its strict subgraphs are CSPs. For CSPs, only the minimal CSPs (MCSPs) are discovered (Ting et al. 2006).

2.3 Distinguishing Subgraph Pattern

The distinguishing subgraph pattern (DSP) was proposed by Zeng et al. (2008). Given two graph datasets D_P , D_N and two support thresholds α , β ($\alpha, \beta \in [0, 1], \alpha \gg \beta$, where \gg means much greater than), a subgraph g is a DSP if $\text{supp}_P(g) \geq \alpha$ and $\text{supp}_N(g) \leq \beta$. The pattern g is a minimal DSP (MDSP) if no strict subgraphs of g are DSPs. Among all DSPs, only MDSPs are discovered (Zeng et al. 2008).

2.4 Analysis

To illustrate and analyse the above patterns, in Fig.1 we use a plane rectangular coordinate system (similar to the support plane (Dong et al. 1999)) to represent any ESP g and its growth rate ($GR(g) = tg(\delta)$). The closer to line OD a point g is, the higher $GR(g)$ is, i.e., the more discriminative the ESP g is. In Fig.1, ESPs are the points in the triangle OGD , CSPs are the points on the line OD , and DSPs are the points in the rectangle $ABCD$.

The CSPs are the most discriminative ESPs with infinite growth rates. However, some problems occur when they are applied to real datasets. First, the constraint is so strict that sometimes no or only a few such patterns exist in the datasets. Second, CSPs are so sensitive to noise that false patterns could be involved in the result and some real patterns could be missed when the patterns are corrupted by noise. For example, (1) if a noise \circ (a vertex or an edge) appears at least one time in D_P and never appear in D_N , then \circ will be found as a MCSP; (2) assumed that $g' = g \diamond e$ (g' is extended from g by adding an edge e) is a real MCSP, and g appears in D_N , if e is added to one of the matches of g in D_N by mistake, then g' will be missed. In addition, the CSP is a kind of discriminative multiple subgraph pattern since disconnected graphs are permitted. This disconnectness allowance blows up the search space (Ting et al. 2006) and makes it not applicable to the scenario of individual substructure analysis.

With the thresholds α and β , the DSP is not so sensitive to the noise with low frequencies². However, to obtain significantly discriminative patterns, α is

¹Both connected subgraphs and disconnected subgraphs are permitted

²The frequencies of the noise are assumed to be lower than the support threshold in this paper

usually needed to be specified a very high value and β a very low value. With this specification, discriminative patterns in the quadrangle $ABHO$ in Fig.1 will be missed. Another drawback of MDSP is that some more discriminative patterns could be missed as MDSPs are not necessarily the most discriminative. For example, if g is a MDSP, then all super-patterns of g will not be included in the result. Thus, more discriminative patterns (g 's super-patterns with higher GR values) will be missed.

Another choice for discriminative subgraph patterns is a complete set of emerging subgraph patterns. However, it is not practicable as finding all ESPs is of high time-complexity, and in real applications, usually users are only interested in the k most discriminative patterns rather than all of them. In Fig. 1 the real top- k most discriminative patterns are the k black circles in region R (between line OD and the dotted curve) with green crosses (false patterns corrupted by noise) and white triangles (redundant patterns) filtered. However, as shown in Fig.1, both CSPs and DSPs only include part of the black circles. Additionally, as analysed above, the discovered MDSPs and MCSPs could be inaccurate with the risk of missing highly discriminative patterns and containing false patterns in the result. Moreover, redundant patterns are not considered and filtered in both CSP and DSP.

3 Motivations

As analysed above, none of the existing patterns, ESPs, MCSPs and MDSPs, include the top- k most discriminative subgraph patterns exactly, and no approaches have been proposed for mining top- k discriminative subgraph patterns. Therefore, it is necessary to introduce a more discriminative and more accurate pattern, and devise an efficient algorithm for the discovery of top- k such patterns.

Furthermore, we identify that none of the existing patterns include the information of patterns' structure changes and discriminating power changes. In substructure analysis, this change information is important. For example, a commonly used principle in chemistry and medicine domains is that structurally similar compounds are more likely to exhibit similar properties (Bender et al. 2004). The principle reflected by grow rates is that structurally similar subgraphs have comparative grow rates. An exception of the principle is that two structurally similar compounds exhibit different properties, i.e., the difference between their growth rates is very big. These two classes (normal and exceptional) of change information are interesting and significant for exploring a pattern's structure change and its impact on the property. In our new discriminative subgraph pattern, the two classes of change information are represented by two subpatterns called "base" and "breaker" respectively. The basic idea is illustrated by an example as follows.

Example 1 Given two graph datasets D_P (Fig.2(a)) and D_N (Fig.2(b)) which consist of molecular structures of two contrasting classes of compounds respectively, assume that the compounds in D_P exhibit a positive property (e.g., toxicity) and the compounds in D_N exhibit the corresponding negative property (e.g., non-toxicity). The vertex labels X , Y and Z are abstract representations of concrete atoms, and the implicit vertex labels in the rings correspond to atom C (carbon). Given $Min_GR = 2.0$, the discovered top-1 ESP is g_1 in Fig.2(c). We examine the structure change and growth rate change of the patterns in Fig.2(c) (growth rates are in the brackets).

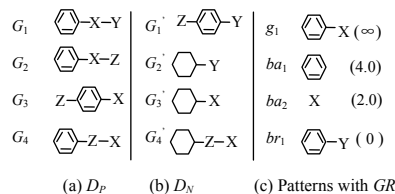


Figure 2: Graph datasets and subgraph patterns

The pattern g_1 indicates that compounds containing g_1 are likely to exhibit the positive property. We examine g_1 's structure changes and growth rate changes. Considering ba_1 and ba_2 in Fig.2 (c), we notice that ba_1 and ba_2 are subgraphs of g_1 , and they are two minimal ESPs, i.e., there exist no subgraphs of ba_1 and ba_2 that are ESPs. Intuitively ba_1 and ba_2 can be seen as two "bases" of g_1 . Then we consider br_1 . The pattern br_1 is structurally similar to g_1 since br_1 can be formed from g_1 by replacing the atom "X" with "Y". We notice that g_1 's growth rate decreases sharply from ∞ to 0. The br_1 appears in D_N but never appears in D_P . This indicates that g_1 loses the positive property and exhibits strong negative property after being "broken" by replacing the atom "X" with "Y". The pair of br_1 and the operation can be seen as a "breaker" of g_1 . For experts this information is not only useful for exploring the inner molecular mechanism of the property but also helpful for finding out ways to weaken or remove the property.

It is obvious that the ESPs with the "bases" and "breakers" are more informative. This type of ESP is called breaker ESP in this paper. This paper aims at defining the breaker ESP, and proposing an efficient approach to discover top- k breaker ESPs.

4 Breaker Emerging Subgraph Pattern

In this section, a new type of discriminative subgraph pattern, breaker ESP (BESP) is defined. After the definition of preliminary concepts, three constraints are introduced into ESPs; then the two subpatterns, base and breaker, are introduced; finally the BESP is defined.

4.1 Preliminary Concepts

The graphs considered in this paper are undirected labeled graphs.

Definition 1 (Undirected Labeled Graphs)

An undirected labeled graph G can be represented by a 5-tuple, $G = \{V, E, \Sigma_V, \Sigma_E, \lambda\}$, where V is a nonempty set of vertices, $E \subseteq V \times V$ is a set of undirected edges, Σ_V and Σ_E are the sets of vertex labels and edge labels respectively. The function λ defines the mappings from vertices to vertex labels, $V \rightarrow \Sigma_V$, and from edges to edge labels, $E \rightarrow \Sigma_E$.

Definition 2 (Subgraphs) G is a subgraph of G' (denoted by $G \subseteq G'$) iff $V \subseteq V'$ and $E \subseteq E' \cap (V \times V)$. $G \subset G'$ denotes G is a strict subgraph of G' .

Definition 3 ((Sub)Graph Isomorphism)

Graph $G = \{V, E, \Sigma_V, \Sigma_E, \lambda\}$ is graph isomorphic to another graph $G' = \{V', E', \Sigma'_V, \Sigma'_E, \lambda'\}$ iff there exists a bijection $f : V \rightarrow V'$ such that for $\forall u \in V, f(u) \in V'$ and $\lambda(u) = \lambda'(f(u))$, and for $\forall e = (u, v) \in E, e' = (f(u), f(v)) \in E'$ and $\lambda(e) = \lambda'(e')$. Graph G is subgraph isomorphic to G' if there exists a subgraph G'' of G' such that G is graph isomorphic to G'' .

Definition 4 (Growth Rate of a Subgraph)

Given two graph datasets D_P and D_N , the growth rate of a subgraph g from D_N to D_P , $GR_{D_N \rightarrow D_P}(g)$, is defined as equation (1) (X is replaced by g).

Definition 5 (Emerging Subgraph Patterns)

Given two graph datasets D_P , D_N and a threshold of growth rate, Min_GR , a set of emerging subgraph patterns (ESPs) from D_N to D_P is defined as:

$$ESP_{D_N \rightarrow D_P} = \{g | GR_{D_N \rightarrow D_P}(g) \geq Min_GR\} \quad (2)$$

It should be remarked that in the rest of the paper, the subscript $D_N \rightarrow D_P$ is omitted when it is apparent, i.e., $GR(g) = GR_{D_N \rightarrow D_P}(g)$, $Min_GR = Min_GR_{D_N \rightarrow D_P}$ and $ESP = ESP_{D_N \rightarrow D_P}$. Similarly, $GR_{D_P \rightarrow D_N}$ and $ESP_{D_P \rightarrow D_N}$ can be defined. For convenience, in the rest of the paper, the subscript “ N ” is used to denote “ $D_P \rightarrow D_N$ ” instead, i.e., $GR_N(g) = GR_{D_P \rightarrow D_N}(g)$, $Min_GR_N = Min_GR_{D_P \rightarrow D_N}$ and $ESP_N = ESP_{D_P \rightarrow D_N}$.

Definition 6 ((Maximum) Common Subgraph)

A common subgraph of G_1 and G_2 is a graph G such that there exist subgraph isomorphism from G to G_1 and from G to G_2 . We call G a maximum common subgraph of G_1 and G_2 , $MCS(G_1, G_2)$, if there exists no other subgraph of G_1 and G_2 that has more vertices than G (Wang et al. 2005).

4.2 Three Constraints on ESPs

As analysed in Section 2.4, the existing patterns, ESPs, MCSPs and MDSPs, have some drawbacks. The constraints α and β for MDSPs lead to the risk of missing highly discriminative patterns. Both ESPs and MCSPs have no constraints on support. This leads to two problems: one is huge searching complexity and the other is the inaccuracy due to the noise of low frequencies. Additionally, redundant patterns are not filtered in the three patterns. To overcome these drawbacks, we exert three constraints on ESPs.

The first constraint is the minimal support threshold. In our definition, any ESP g must be frequent, i.e., $sup_P(g) \geq min_sup_P$, where min_sup_P is the threshold of sup_P . This constraint brings three advantages: (1) it ensures that the patterns are popular to some degree in the dataset; (2) it filters the noise with low frequencies, and thus filters some false patterns corrupted by the noise; (3) it greatly reduces the number of patterns that need to be generated, and thus reduces the computational complexity.

The second constraint is that any ESP g must be closed in D_P , that is to say there exist no proper supergraphs of g that have the same support of g . The first reason for exerting this constraint is that it can further reduce the number of patterns that need to be generated. The second reason is that it can filter some redundant patterns without losing significant ESPs. For example, if g is not closed, i.e., $\exists g'$ such that $g \subset g'$ and $sup_P(g) = sup_P(g')$, then $GR(g) \leq GR(g')$ since $sup_N(g) \geq sup_N(g')$. Therefore, for g' , g is redundant. After adding the constraint, g will be pruned.

The third constraint is for pruning redundant patterns that have subgraphs or super-graphs with higher growth rate values. For a pair of ESPs g and g' having the relationship: $g \subset g'$ (or $g' \subset g$), (1) if $GR_P(g) > GR_P(g')$ then g' is pruned as a redundant pattern; (2) if $GR(g) = GR(g')$ and $sup_P(g) \neq sup_P(g')$ then the larger graph is pruned as a redundant pattern.

Given D_P , D_N , min_sup_P and Min_GR , the ESPs that satisfy the min_sup_P constraint is called frequent ESPs (FESPs), and the ESPs that satisfy the first

two constraints above are called closed frequent ESPs (CFESPs), and the ESPs that satisfy the three constraints are called constrained ESPs (CESPs). The set of FESPs, CFESPs and CESPs are denoted by $FESP$, $CFESP$ and $CESP$ respectively.

4.3 Breaker Emerging Subgraph Pattern

As indicated in Example 1, besides each ESP itself, the bases and breakers of the pattern should be provided. A breaker ESP consists of three subpatterns: a CESP, a set of bases and a set of breakers.

The bases of a CESP g_i are defined as the minimal CFESPs in the subgraphs of g_i , which are formally defined below.

Definition 7 (The bases of a CESP) Given D_P , D_N , min_sup_P and a set of CESPs, $CESP = \{g_i\}$, for $\forall g_i \in CESP$, the set of bases of g_i , Ba_i , is defined as

$$Ba_i = \{ba | ba \in CFESP, ba \subset g_i, \text{ and } \neg \exists s \in CFESP \text{ such that } s \subset ba\} \quad (3)$$

As shown in Example 1, a breaker pattern, br , of a CESP g_i is structurally similar to g_i , but its growth rate decreases significantly. Two types of breakers are interesting. One is that br still appears frequently in D_P , but its growth rate is weakened to a value below Min_GR . The other is that br appears more frequently in D_N than in D_P , i.e., $GR_N(br) > 1$. The first type exhibits the same property as g_i to some extent, while the second type exhibits the opposite property. The first type is called weakening breaker, and the second type is called reverse breaker. To define the breakers, two metrics are needed to measure the structural similarity and the change degree of growth rate.

For real graph data from different applications, the standards for measuring the structural similarity could vary. Even in the same domain such as chemistry, dozens of similarity coefficients are available for measuring the structural similarity (Nikolova et al. 2004). In this paper, we just adopt a commonly used metric, the maximum common subgraph, to measure the structural similarity. The similarity degree between g_i and a candidate breaker pattern br is quantified by:

$$Similarity(g_i, br) = \frac{2|MCS(g_i, br)|}{|g_i| + |br|} \quad (4)$$

where, $|g_i|$ refers to the size of g_i . Two patterns are structurally similar if their $Similarity$ is no less than a user specified threshold $\delta \in (0, 1)$.

The graph size can be evaluated by edge number or vertex number. The $Similarity$ is denoted by $Similarity1$ ($Similarity2$) when vertex (edge) number is used.

For the first type of breaker, the change degree of the growth rate of a breaker candidate, br , of g_i , can be defined as

$$GR_change(g_i, br) = \begin{cases} \infty & \text{if } GR(br) = 0 \\ \frac{GR(g_i)}{GR(br)} & \text{otherwise} \end{cases} \quad (5)$$

The change degree is significant if $GR_change(g_i, br)$ is no less than a user specified threshold $\rho > 1$.

For the second type of breaker, GR_N represents the change degree, i.e., the degree that a breaker pattern exhibits the negative property.

The two types of breakers are formally defined as follows.

Definition 8 (Weakening Breaker) Given D_P , D_N , $CESP = \{g_i\}$, $CFESP$, Min_GR , δ and ρ , for $\forall g_i \in CESP$, the set of weakening breakers of g_i , WBr_i , is defined as

$$WBr_i = \{\langle br, \varphi \rangle | br \in CFESP, br = \varphi(g_i), 1 \leq GR(br) < Min_GR, Similarity(g_i, br) \geq \delta, GR_change(g_i, br) \geq \rho\} \quad (6)$$

A breaker of g_i consists of a breaker pattern br and a breaker operator φ , which is a set of operations that transforms g_i to br , and $br = \varphi(g_i)$ means that br can be formed by conducting the operator φ on g_i . The operations in φ are from 6 basic operations on graphs: AV (adding a vertex), AE (adding an edge), DV (deleting a vertex), DE (deleting an edge), MV (modifying a vertex label) and ME (modifying an edge label). Since the bases reflect the information of the patterns with GR no less than Min_GR , $GR(br)$ is constrained to be less than Min_GR .

Definition 9 (Reverse Breaker) Given D_P , D_N , $CESP = \{g_i\}$, min_sup_N , Min_GR_N and δ for $\forall g_i \in CESP$, the set of reverse breakers of g_i , RBr_i , is defined as

$$RBr_i = \{\langle br, \varphi \rangle | br = \varphi(g_i), Similarity(g_i, br) \geq \delta, sup_N(br) \geq min_sup_N, GR_N(br) \geq Min_GR_N\} \quad (7)$$

The threshold min_sup_N is used to ensure that br is popular to some extent in D_N and to filter the noise with low frequencies in D_N . The Min_GR_N is used to ensure that br loses the positive property and exhibits the negative property to some degree.

Based on the CESP and the definitions of base and breaker, breaker ESPs (BESPs) are defined as follows.

Definition 10 (Breaker ESPs) Given D_P , D_N , min_sup_P , min_sup_N , Min_GR , Min_GR_N , δ and ρ , the set of BESPs from D_N to D_P is defined as

$$BESP = \{\langle g_i, Ba_i, Br_i \rangle | g_i \in CESP\} \quad (8)$$

where, Ba_i is the set of bases of g_i and Br_i is the set of breakers of g_i ($Br_i = WBr_i \cup RBr_i$).

A breaker ESP is composed of three subpatterns: a constrained ESP, g_i , a set of bases Ba_i of g_i , and a set of breakers Br_i of g_i . It should be noted that Ba_i (Br_i) is an empty set when no bases (breakers) of g_i exist in the datasets. In implementation, the BESPs are sorted by the growth rate of g_i in descending order, and only the top-k BESPs are discovered, where k is a user-specified integer.

5 Mining Top-k Breaker Emerging Subgraph Patterns

An efficient algorithm for mining top-k BESPs, k-MBESP, is proposed in this section. The top-k BESPs are discovered in three main stages:

1. Find top-k constrained ESPs, $CESP_K = \{g_1, \dots, g_k\}$;
2. Find the bases of each $g_i \in CESP_K$;
3. Find the breakers of each $g_i \in CESP_K$.

5.1 Finding Top-k Constrained ESPs

The top-k CESP are found by 4 steps. First, the set of closed frequent subgraphs, CF , is discovered from D_P . Second, all closed frequent subgraphs are inserted into a layered graph L as shown in Fig.3.

Algorithm 1 Top-k-CESP

Comments: find top-k CESP.

Input: D_P , D_N , min_sup_P , Min_GR , k

Output: Top-k CESP, $CESP_K$

- 1: $CESP_K \leftarrow \emptyset$;
- 2: Scan D_P once to find frequent vertices FV ;
- 3: **for** each vertex $v \in FV$ **do**
- 4: CloseGraph(v , $NULL$, D_P , min_sup_P , CF);
- 5: **for** each graph G_i' in D_N **do**
- 6: GR-Computation(L , G_i' , $|D_P|$, $|D_N|$, Min_GR);
- 7: Traverse L to single out $CESP_K$;

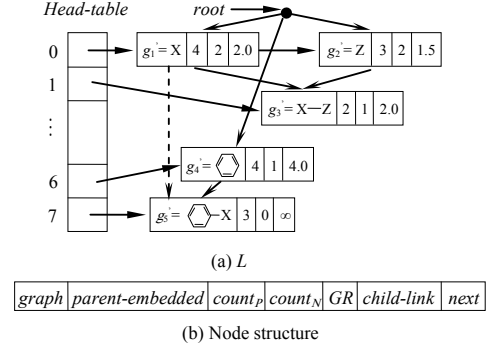


Figure 3: The layered graph L and its node structure

Third, D_N is scanned once to compute sup_N and GR of each subgraph in L . Finally, the top-k CESP are detected and output from L . The procedure is described in Algorithm 1.

In Line 4, the CloseGraph algorithm (Yan et al. 2003) is adopted to find CF first. In the implementation of CloseGraph, an additional subprocedure is added to insert each found frequent closed subgraph into L as shown in Fig.3(a). In Line 6, GR-Computation computes current $count_N$ and GR of each node in L . The following example is used to illustrate the algorithm.

Example 2 Given D_P and D_N as shown in Fig.2(a)(b), let $min_sup_P=0.5$, $Min_GR=2.0$, $min_sup_N=0.25$, $Min_GR_N=2.0$, $\delta = 0.8$, $\rho = 10$ and $k = 2$. The k-MBESP algorithm is used to discover top-2 BESPs.

In Example 2, firstly, $FV = \{C, X, Z\}$ is found; then CloseGraph generates $CF = \{g_1', g_2', \dots, g_5'\}$, and inserts each $g_i' \in CF$ into L as shown in Fig.3(a). Each g_i' and its related information are stored in a node of L . For space limitation, only 4 domains ($graph$, $count_P$, $count_N$ and GR) are shown explicitly. The node structure is shown in Fig.3(b). The L is organised in two dimensions. Horizontally, the graphs with the same edge number are organised in the same layer. Level numbers (edge numbers) are stored in the Head-table. Vertically, a child-link is set from a parent to a child (a node c is a child of node p if $p.graph \subset c.graph$). The root is chosen as its parent when a node has no parents. To avoid generating too many links, child-links are only set from the nearest parents to the children, e.g., a child-link is not set from g_1' to g_5' .

The GR-Computation procedure is described in Algorithm 2. The basic idea is for each graph G_i' in D_N , to search L from top to down to test whether the subgraph in each node is embedded in G_i' . To reduce the time complexity of subgraph-isomorphism test, two pruning strategies are introduced.

- **Pruning strategy 1** For node u in Level l , if $u.graph \not\subseteq G_i'$, then u 's children are pruned

Algorithm 2 GR-Computation

Comments: compute $count_N$ and GR of each node in L after G_i' is scanned.

Input: L , G_i' , $|D_P|$, $|D_N|$ and Min_GR

Output: L in which $count_N$ and GR of each node have been computed after G_i' is scanned

```

1: for each node  $u \in L$  do
2:    $u.parent\_embedded \leftarrow true$ ;
3:   for  $l=0$  to  $max\_layer$  do
4:      $u \leftarrow L.Head\_table[l]$ ;
5:     while  $u \neq null$  do
6:       if  $u.GR \neq -1$  then
7:         if  $u.parent\_embedded = true$  and
            $u.graph \subseteq G_i'$  then
8:            $u.count_N \leftarrow u.count_N + 1$ ;
9:            $u.GR \leftarrow \frac{u.count_P/|D_P|}{u.count_N/|D_N|}$ ;
10:          if  $u.GR < Min\_GR$  then
11:             $u.GR \leftarrow -1$ ;
12:          else for each child  $c$  of  $u$  do
13:             $c.parent\_embedded \leftarrow false$ ;
14:          if  $i = |D_N|$  and  $u.count_N = 0$  then
15:             $u.GR \leftarrow \infty$ ;
16:           $u \leftarrow u.next$ ;
```

Algorithm 3 Base-Detection

Comments: find the bases of each $g_i \in CESP_K$.

Input: L and $CESP_K$

Output: $Ba_i (i = 1, 2, \dots, k)$

```

1: for each  $Ba_i$  do  $Ba_i \leftarrow \emptyset$ ;
2: for each child  $u$  of  $L.root$ 
3:   if  $u.GR \neq -1$  then
4:     for each  $g_i \in CESP_K$ 
5:       if  $u.graph \subset g_i$  then
6:          $Ba_i \leftarrow Ba_i \cup \{u.graph\}$ ;
```

(Note: this pruning does not mean really pruning the nodes from L but means that subgraph-isomorphism test need not be conducted between any graphs in the nodes and G_i').

- **Pruning strategy 2** The subgraph-isomorphism test need not be done for node u if current $u.GR$ is less than Min_GR .

Pruning strategy 1 is implemented by Line 7, 12 and 13 in Algorithm 2. In Line 7, the second condition is tested only if the first condition is true. Pruning strategy 2 is implemented by Line 6, 10 and 11. A special value of -1 is used to indicate that current $u.GR$ is less than Min_GR . In Example 2, $count_N$ and GR of each node in L are computed by GR-Computation and their values are shown in Fig.3(a).

The last step of the Top-k-CESP procedure is, based on GR and the third constraint, traversing L and identifying top-2 CESP, $CESP_2 = \{g_1 = g_5' : \infty, g_2 = g_3' : 2.0\}$.

5.2 Finding the Bases

The bases of each g_i in $CESP_K$ can be found easily from L since they are kept in L . Note that if there is a child-link from the root to node u , then $u.graph$ is a minimal CFESP. Therefore, based on Definition 7, if $u.graph \subset g_i$, then $u.graph$ is a base of g_i . The procedure is described in Algorithm 3. In Example 2, for g_1 , $Ba_1 = \{g_1', g_4'\}$, and for g_2 , $Ba_2 = \{g_1'\}$.

Algorithm 4 WBreaker-Identification

Comments: find the weakening breakers of each $g_i \in CESP_K$ and reverse breakers in $CFESP$

Input: $|D_P|$, $|D_N|$, $CESP_K$, L , δ , ρ , min_sup_N , Min_GR_N and k

Output: WBr_i , and RBr_i if there exist reverse breaker patterns in $CFESP$ ($i=1,2,\dots,k$).

```

1: for  $i=1$  to  $k$ 
2:   Calculate  $E_{min}(g_i)$  and  $E_{max}(g_i)$ ;
3:   Locate node  $u$  in  $L$  s.t.  $u.graph = g_i$ ;
4:   for each node  $p$  from Layer  $E_{min}(g_i)$  to
       Layer  $E_{max}(g_i)$ 
5:     if  $p.GR = -1$  and ((there exists a path
       from  $p$  to  $u$  or from  $u$  to  $p$ ) or ( $p$  and  $u$ 
       have a common antecedent node from
       Layer  $E_{min}(g_i)$  to Layer  $|g_i| - 1$ )) then
6:        $brC_i \leftarrow brC_i \cup \{p.graph\}$ ;
7:   Scan  $D_N$  to compute  $count_N$  and  $GR$  of each
        $br \in brC$  ( $brC = \bigcup_{i=1}^k brC_i$ );
8:   for each  $g_i \in CESP_K$ 
9:     for each  $br \in brC_i$ 
10:      if  $GR(br) \geq 1$  and  $GR\_change(g_i, br) \geq \rho$ 
         then
11:         $WBr_i \leftarrow WBr_i \cup \{br, \varphi\}$ ;
12:      else if  $GR(br) < 1$ ,  $1/GR(br) \geq$ 
          $Min\_GR_N$ ,  $sup_N(br) \geq min\_sup_N$  then
13:         $RBr_i \leftarrow RBr_i \cup \{br, \varphi\}$ ;
```

Algorithm 5 RBreaker-Identification

Comments: find the reverse breakers of each $g_i \in CESP_K$.

Input: $|D_P|$, $|D_N|$, $CESP_K$, min_sup_N , Min_GR_N , δ , k

Output: $RBr_i (i = 1, 2, \dots, k)$

```

1: Find  $FESP_N$  using Algorithm 1;
2: for  $i=1$  to  $k$ 
3:   for each  $br \in FESP_N$ 
4:     if  $Similarity1(g_i, br) \geq \delta$  then
5:        $RBr_i \leftarrow RBr_i \cup \{br, \varphi\}$ ;
```

5.3 Finding the Breakers

Two breaker identification procedures are devised for the discovery of two types of breakers respectively.

5.3.1 Finding the Weakening Breakers

The procedure is described in Algorithm 4. In this procedure, edge number is used to evaluate graph size. First, obtain the minimum (maximum) edge number, $E_{min}(g_i)$ ($E_{max}(g_i)$) of candidate breaker patterns of each $g_i \in CESP_K$. Given δ and $|g_i|$, $E_{min}(g_i)$ and $E_{max}(g_i)$ can be derived easily from Equation (4) according to $|MCS(g_i, br)| \leq \min\{|g_i|, |br|\}$. Second, traverse L from Layer $E_{min}(g_i)$ to $E_{max}(g_i)$ to detect candidate breaker patterns, brC_i , of g_i (Line 4, 5 and 6). Line 5 identifies the candidates that satisfy thresholds Min_GR and δ in Equation (6). Third, scan D_N to calculate $count_N$ and GR of each candidate. Finally check if the candidates satisfy the threshold ρ (Line 10). If there exists such br that satisfies the constraints of reverse breakers, then br is inserted into the corresponding reverse breaker set (Line 12 and 13).

5.3.2 Finding the Reverse Breakers

The procedure is described in Algorithm 5. In Line 1, the constraints $closed$ and k are not needed in Algorithm 1 for finding frequent ESPs from D_P to D_N ,

$FESP_N$. The key of the computation of $Similarity1$ (Equation (4)) in Line 4 is to identify the maximum common subgraph of g_i and br , $MCS(g_i, br)$. An existing efficient algorithm (Wang et al. 2005) is implemented to detect the minimal common subgraphs. The $MCS(g_i, br)$ is discovered by five major steps: (1) produce the matching pairs of two input graphs; (2) sort the order of matching pairs; (3) build a common subgraph path through selecting matching pairs; (4) determine the size of the corresponding common subgraph by the path; (5) continue finding paths until all paths have been considered. In $FESP_N$ usually almost no or only a small number of candidates are structurally similar to g_i . In order to accelerate the procedure, two pruning strategies are introduced to prune the search space.

- **MCS-pruning 1** If $\frac{2\min\{|g_i|, |br|\}}{|g_i| + |br|} < \delta$ then $Similarity1(g_i, br) < \delta$ because $|MCS(g_i, br)| \leq \min\{|g_i|, |br|\}$. In this case, $MCS(g_i, br)$ need not be identified.
- **MCS-pruning 2** In Step (1) of the MCS detection procedure, if the number of matching pairs, NMP , does not satisfy $\frac{2NMP}{|g_i| + |br|} \geq \delta$, then retreat from the procedure. Also in Step (3) only consider the pathes whose NMP satisfy $\frac{2NMP}{|g_i| + |br|} \geq \delta$.

In Example 2, the discovered reverse breakers are $Br_1 = \{\langle br_1, MV(v_7, Y) \rangle\}$, where br_1 is from Fig.2(c), and $MV(v_7, Y)$ means the label of v_7 (the vertex with label "X" in br_1) is modified to Y .

6 Experimental Results and Analysis

To evaluate the k-MBESP algorithm, experiments were conducted on both real and synthetic datasets. All experiments were done on a 2.2GHz Intel Core PC, with 2 GB main memory, running Windows XP. For comparison, we also implemented the algorithm for mining MCSPs (Ting et al. 2006) and the algorithm for mining MDSPs (Zeng et al. 2008), which are denoted by MCSP-Miner and MDSP-Miner respectively. All algorithms were implemented in Java.

6.1 Real Dataset

The real dataset that we use is the AIDS antiviral screen chemical compound dataset obtained from the website³. The dataset contains 42687 compounds, among which, 377 are confirmed active (CA), 1911 are confirmed moderately active (CM) and 40389 are confirmed inactive (CI). In the experiments, we only focus on CA and CI compounds. CA compounds are stored in D_P and CI compounds are stored in D_N . The k-MBESP algorithm is used to find the top-k BESPs from D_N to D_P . We determine an appropriate specification for the parameters: $min_sup_P=14\%$, $min_sup_N=14\%$, $k=10$, $Min_GR=25.0$, $Min_GR_N=3.0$, $\delta=0.65$, $\rho=8.0$. With this specification, the algorithm finds top-10 BESPs within 2835 seconds. The top-1,2 and 7 BESPs are shown in Fig.4. The value in every bracket is the GR of a pattern. Several bases are found for each g_i , among which only a couple of them are shown in Fig.4. For g_7 , a weakening breaker pattern, wbr_{7-1} , is discovered. We see that the growth rate of g_7 decreases from 88.76 to 8.82 ($GR_change=10.06$) when three bonds attached to atom S are broken as shown by the three dashed line in Fig.4. This information

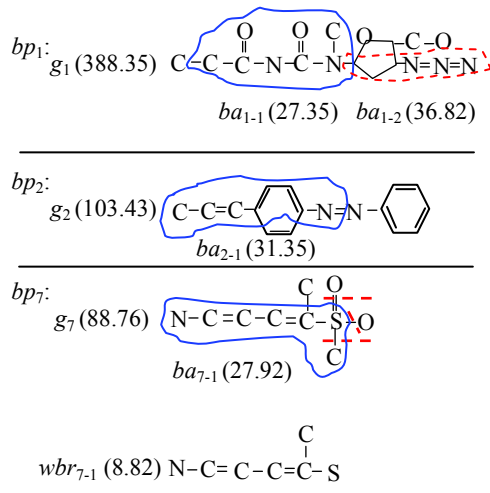


Figure 4: top-k BESPs discovered from the real dataset

is heuristic and important for domain experts to discover the factors that could weaken the activity of the compounds. It should be noted that breaker patterns are not necessarily underlying in the datasets. No breaker patterns can be found if no breaker patterns exist in the datasets to be mined. For example, no reverse breakers are found in the AIDS dataset when the parameters are specified as above.

In this dataset, MDSP-Miner ($\alpha=14\%$, $\alpha/\beta=Min_GR=25$) only finds the minimum ESPs, and misses some more discriminative patterns. For example, some bases of g_1 , g_2 are found as MDSPs and g_1 , g_2 are excluded. However, MCSP-Miner is not able to finish the mining process within an acceptable time.

6.2 Synthetic Datasets

In order to evaluate the performance of the algorithm, we generated a series of synthetic datasets by a synthetic graph generator (Kuramochi et al. 2001) with fixed parameters I5T20L200V6E4 and varying D, where D denotes the average size of frequent patterns (in terms of edge number), T denotes the average size of graph transactions, L denotes the number of potentially frequent subgraphs, V denotes the number of distinct vertex labels, E denotes the number of distinct edge labels, and D denotes dataset size (the number of graphs in the dataset).

6.2.1 Performance Study

To compare the time efficiency, the three miners are performed on a series of datasets with the size varying from 20 to 100k. The parameters for k-MBESP are specified as: $Min_GR=25.0$, $min_sup_P=5\%$, δ (at most 2 vertices or edges are different), $\rho=30.0$, $Min_GR_N=10.0$, $min_sup_N=5\%$ and $k=10$. For MDSP-Miner, the parameter $\alpha=5\%$, and the value of α/β is fixed at 25.0, which is as same as Min_GR . However, when the dataset size is 20 or 100, $Min_GR=5.0$, $min_sup_P=10\%$, $\alpha=30\%$, and $\beta=6\%$. Figure 5(a) shows a performance comparison of the three miners on datasets of 20 to 10000 graphs. As shown in Fig.5(a), k-MBESP is more efficient than two previous miners. When the dataset size is over 10k, both MCSP-Miner and MDSP-Miner are not able to finish the mining process in 3 hours. In contrast, k-MBESP can finish within 1000 seconds even on large datasets of up to 100k graphs. Fig.5(b) shows the runtime when $D=5k$ and min_sup_P for k-MBESP (α for MDSP-Miner) varies from 1% to

³http://dtp.nci.nih.gov/docs/aids/aids_data.html

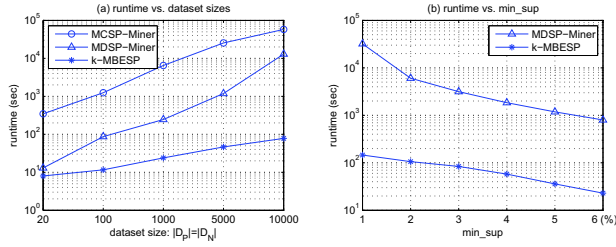


Figure 5: A performance comparison of the three miners

6%. We see that k-MBESP has better scalability on min_sup than MDSP-Miner.

The high efficiency and good scalability of our algorithm benefit from the constraints, the compact data structure and the pruning strategies that we introduced. Firstly, the min_sup and $closed$ constraints greatly reduce the search space of subgraph candidates. Secondly, the high compact layered graph contributes to the high efficiency. All candidates are stored in the layered graph which can be loaded into main memory prior to the computation of GR . Thus only one scan of the datasets is needed to compute GR for all candidates. In contrast, a large number of scans are required in the other two miners. In MDSP-Miner, one scan of the dataset is needed for each MDSP candidate, therefore the minimal number of scans is the number of MDSP candidates. In MCSP-Miner, for each graph in D_P , one scan of D_N is required for discovering the maximal common edge sets (Ting et al. 2006). consequently, for a dataset of 5k graphs, at least 5k scans are needed for MCSP-Miner. Thirdly, the pruning strategies further reduce the time complexity.

6.2.2 A Comparison of Discovered Patterns

We also compare the patterns discovered by the three miners to evaluate their informativeness, accuracy and discriminating power. Figure 6(a)(b)(c) show the patterns discovered by the three miners from datasets D5I5T20L200V6E4, which are denoted by bp_i , cp_i and dp_i respectively. In Fig.6(a) ba_{i-j} denotes the j^{th} base of g_i . The patterns in Fig.6(a)(c) are sorted according to GR values of CESPs and MDSPs respectively in descending order. As shown in Fig.6(a), for g_1 and g_2 , two weakening breakers, wbr_{1-1} and wbr_{2-1} , are discovered, and both GR_change values are ∞ . For g_4 , a reverse breaker, rbr_{4-1} , is discovered and $GR_N = 12.8$. This indicates that g_4 loses the strong ($GR = \infty$) positive property and exhibits the negative property to some extent ($GR_N = 12.8$).

It is obvious that the top-k BESPs are more informative with the information of CESPs and their bases and breakers. Comparing bp_1 to cp_1 and dp_1 in Fig.6, we see that the CESP g_1 in bp_1 is just cp_1 and dp_1 . The information of MCSPs (except disconnected subgraphs) and MDSPs are included in BESPs.

To test the accuracy of the miners, we introduce two noises in Fig.6(d) into the datasets: (1) a noi_1 is added into D_P , (2) a noi_2 is introduced into D_N by modifying a edge label of a subgraph in D_N from e to f . The result of k-MBESP is not affected by noi_1 , and for bp_1 , g_1 's growth rate is changed to $count_P(g_1)=1364$. However, the noi_1 is found as a MCSP, cp_1^* , by MCSP-Miner since $GR(noi_1) = 1/0 = \infty$, and when noi_2 appears, cp_1 , i.e., g_1 , can not be found by MCSP-Miner since $GR(cp_1) \neq \infty$. The result of MDSP-Miner is affected as same as that of k-MBESP by noi_1 and noi_2 . In addition, MDSP-

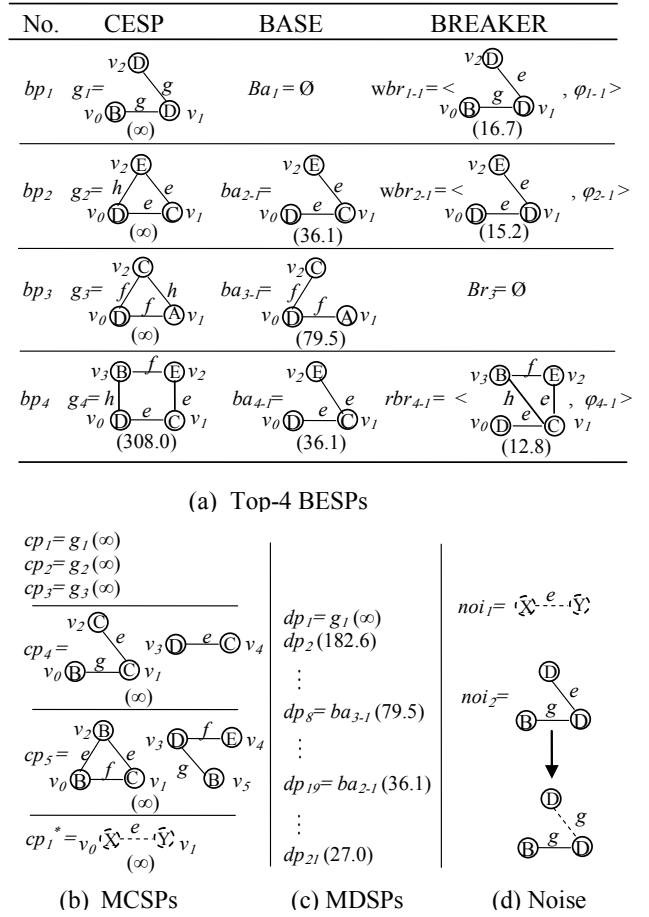


Figure 6: A comparison of patterns discovered by three miners

Miner could miss important patterns in the following two cases. First, as shown in Fig.6(c), g_2 is among the most discriminative patterns with infinite GR , but it is missed by MDSP-Miner since it is replaced by dp_{19} . It is clear that g_2 is more discriminative than dp_{19} , but g_2 is replaced by dp_{19} as $GR(dp_{19}) = 36.1 > 25.0$ and $dp_{19} \subset g_2$. Similarly, g_3 and g_4 are missed. Second, as shown in Fig.5(b), MDSP-Miner can not finish the mining process in an acceptable time when min_supp (i.e., α) is specified a very low value. Therefore, sometimes those MCSPs with low $supp$ could not be found by MDSP-Miner. In contrast, k-MBESP can accept a relatively lower min_supp value. In addition, the constraints in BESPs filters some redundant patterns. Compared with the other two miners, k-MBESP is more accurate as it filters redundant patterns and false patterns corrupted by noise with low frequencies and does not miss more discriminative patterns.

As for discriminating power, the top-k BESPs are the top-k most discriminative patterns in terms of grow rate. In contrast, MDSPs are not necessarily the most discriminative, and some more discriminative patterns could be missed as examined above.

As for mining power, k-MBESP is more powerful than the other two miners. Figure 6(a) shows that k-MBESP is capable of discovering top-k BESPs. The other two miners can not discover them. One exception is that disconnected subgraph patterns such as cp_4 and cp_5 in Fig.6(b) are not considered in our miner since we only aim at detecting individual subgraph patterns of high discriminating power.

Another advantage is that, with relatively small number of patterns, top-k BESPs are more convenient for domain experts to select, examine and analyse. Furthermore, the change information of pattern

structure and growth rate values kept in the bases and breakers provide heuristic information for the experts and help them discover important knowledge. In contrast, a relatively large number of MDSPs are not convenient for manual examination and analysis, and no change information is contained in both MCSPs and MDSPs.

In summary, the discovered top-k breaker emerging patterns are more informative, more discriminative and more accurate than the MCSPs and MDSPs extracted from the same datasets.

7 Conclusions and Future Work

In this paper, we introduced a new type of discriminative subgraph pattern, breaker emerging subgraph pattern, which consists of three important subpatterns: (1) the top-k CESP that reflect the top-k most significant individual structural differences between two classes of graphs, (2) the bases that indicate structural bases of the discriminative patterns, and (3) the breakers that indicate triggers to weaken the growth rates of the patterns. We also proposed an efficient miner, k-MBESP, for the discovery of top-k BE-SPs. The experimental results show that the miner is capable of finding the top-k BE-SPs efficiently, more efficient, more powerful and more accurate than two previous miners. Compared with the complete sets of MCSPs and MDSPs discovered by previous miners, the top-k BE-SPs extracted by our algorithm have at least the following 4 advantages: (1) more informative (2) more discriminative in terms of growth rate, (3) more accurate, (4) more convenient and more useful for experts' further examination and analysis.

The BE-SP extends the application of discriminative subgraph patterns. It can be applied to: (1) detecting the difference between two contrasting classes of graphs, (2) exploring the inner structural mechanism of the property of a class of graphs, and (3) helping domain experts to discover ways to activate (strengthen) the desired properties, such as the activity to AIDS, and to break (weaken) the undesired properties, such as the toxicity.

Some future work needs to be done. First, more effective noise filtering strategies should be introduced to enhance the robustness of the top-k-BESP miner on data with various noise. Second, study the applications of the BE-SP. For example, use BE-SPs to detect differences between two contrasting classes of compounds or drugs (eg. high curative effects vs. low curative effects, high toxicity vs. low toxicity), to explore the molecular mechanism and to discover ways to design drugs of high curative effects and low toxicity. Based on the differences of website access behaviors between the males and the females represented by BE-SPs, modify the organization of a website to obtain a male-style website and a female-style one.

References

- Agrawal, R., Imielinski, T. & Swami, A. (1993), Mining association rules between sets of items in large databases, in 'ACM SIGMOD International Conference on Management of Data', Vol. 22, ACM Press, Washington DC, USA, pp. 207–216.
- Bender A. & Glen R. C., (2004), Molecular similarity: a key technique in molecular informatics, *Org. Biomol. Chem.* **2**, 3204–3218.
- Cheng D., Yan Xi., Han J. & Yu P. S. (2008), Direct Discriminative Pattern Mining for Effective Classification, in 'International Conference on Data Engineering', pp. 169–178.
- Dong, G. & Li, J. (1999), Efficient mining of emerging patterns: discovering trends and differences, in 'ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', Vol. 22, ACM Press, Washington DC, USA, pp. 207–216.
- Fan W., Zhang K., Cheng H., Gao J., Yan X., Han J., Yu P. S. & Verscheure O. (2008), Direct mining of discriminative and essential frequent patterns via model-based search tree, in 'ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', ACM Press, Washington DC, USA, pp. 230–238.
- Kuramochi M. & Karypis G. (2001), Frequent subgraph discovery, in 'IEEE International Conference on Data Mining', IEEE Computer Society, California, USA, pp. 313–320.
- McGregor J. J. (1982), Backtrack search algorithms and the maximal common subgraph problem, *Software Practice and Experience* **12**, 23–24.
- Nijssen S. & Kok J. N. (2004), A quickstart in frequent structure mining can make a difference, in 'ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', ACM Press, Washington DC, USA, pp. 647–652.
- Nikolova N. & Jaworska J., (2004), Approaches to measure chemical similarity - A review, *QSAR Comb. Sci.* **22**, 1006–1026.
- Ramamohanarao K., Bailey K. & Fan H. (2006), Efficient mining of contrast patterns and their application to classification, in 'International Conference on Intelligent Sensing and Information Processing', IEEE Computer Society, Washington DC, USA, pp. 39–47.
- Sanfeliu A. & Fu K. S. (1983), A distance measure between attributed relational graphs for pattern recognition, *IEEE Trans. Systems, Man and Cybernetics* **13**, 353–362.
- Ting R. M. H. & Bailey J., (2006), Mining minimal contrast subgraph patterns, in 'SIAM International Conference on Data Mining', Society for Industrial and Applied Mathematics, Philadelphia, USA pp. 639–643.
- Wang y. & Maple C., (2005), A novel efficient algorithm for determining maximum common subgraphs, in 'The International Conference on Information Visualisation', IEEE Computer Society, Washington DC, USA, pp. 657–663.
- Yan X. & Han J., (2003), CloseGraph: mining closed frequent graph patterns, in 'ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', ACM Press, Washington DC, USA, pp. 286–295.
- Zeng Z., Wang J. & Zhou L., (2008), Efficient mining of minimal distinguishing subgraph patterns from graph databases, in 'The Pacific-Asia Conference on Knowledge Discovery and Data Mining', Springer-Verlag, Berlin Heidelberg, pp. 1062–1068.

Edge Evaluation in Bayesian Network Structures

Saaïd Baraty¹

Dan A. Simovici¹

¹ University of Massachusetts Boston
Department of Mathematics and Computer Science,
100 Morrissey Blvd, Boston, Massachusetts 02125, USA
Email: {sbaraty, dsim}@cs.umb.edu

Abstract

We propose a measure for assessing the degree of influence of a set of edges of a Bayesian network on the overall fitness of the network, starting with probability distributions extracted from a data set. Standard fitness measures such as the Cooper-Herskowitz score or the score based on the minimum description length are computationally expensive and do not focus on local modifications of networks. Our approach can be used for simplifying the Bayesian network structures without significant loss of fitness. Experimental work confirms the validity of our approach.

Keywords: Bayesian belief network, Kullback-Leibler divergence, entropy, edge pruning

1 Introduction

The construction of a Bayesian Network Structure from a data set that captures the probabilistic dependencies among the attributes of the data set has been one of the prominent problems among community of uncertainty researchers since early 90s. The problem is particularly challenging due to enormity of number of possible structures for a given collection of data.

Formally, a *Bayesian Belief Network* is a pair $(\mathcal{B}_s, \mathcal{B}_p)$, where \mathcal{B}_s is a DAG (directed acyclic graph) which is commonly referred to as a *Bayesian Network Structure* (BNS), and \mathcal{B}_p is a collection of distributions which quantifies the probabilistic dependencies present in the structure, as we discuss in detail below.

Each node of the BNS corresponds to a random variable; edges represent probabilistic dependencies among these random variables. BNS captures the split of the joint probability of a set of random variables, presented by its nodes, into a product of probabilities of its nodes conditioned upon a set of other nodes, namely the set of its *predecessors* or *parents*.

The set of values (or states) of a random variable Z is referred to as the *domain of Z* , borrowing a term from relational databases. This set is denoted by $\text{Dom}(Z)$.

If a random variable X is a node of \mathcal{B}_s with $\text{Dom}(X) = \{1, \dots, R_X\}$ and set of random variables $\text{Pa}_X = \{Y_1, Y_2, \dots, Y_k\}$ as its set of parents, and if we agree upon some enumeration of set $\text{Dom}(\text{Pa}_X) = \prod_{i=1}^k \text{Dom}(Y_i)$, then we denote by θ_{lj}^X the conditional probability $P(X = l | Y_1 = y_1, Y_2 = y_2, \dots, Y_k = y_k)$, where l is some state of X and (y_1, \dots, y_k) is the j^{th} element of the enumeration. Also, we denote with θ_j^X , the

probability distribution of X conditioned on its set of parents taking on the j^{th} assignment of its domain. \mathcal{B}_p is collection of distributions θ_j^X for all nodes X of \mathcal{B}_s and $1 \leq j \leq |\text{Dom}(\text{Pa}_X)|$.

Several scoring solutions have been proposed for evaluating the fitness of a BNS for representing probabilistic dependencies among attributes of a data set. There are two major approaches: scores based on maximization of the posterior probability of the network structure conditioned upon data, and scores based on MDL (Minimum Description Length) principle.

The first approach was initially introduced in Cooper & Herskovits (1993), where the scoring formula was derived based on a number of assumptions such as assuming that the distribution of tuples $(\theta_{1j}^X, \dots, \theta_{R_X j}^X)$ is uniform for all X and j , or is a Dirichlet distribution. In Heckerman et al. (1995) the Dirichlet distribution assumption was replaced by the *likelihood equivalence* assumption and it was shown that under this new assumption tuples $(\theta_{1j}^X, \dots, \theta_{R_X j}^X)$ obey a Dirichlet distribution.

The second approach is based on the minimum description length principle, introduced in Rissanen (1978), which stipulates that the best model for data is the one that minimizes the combined description length of the model and data. Later, in Lam & Bacchus (1994) this principle was applied to learning a BNS from data. The close relationship between these two approaches was shown in Suzuki (1999).

The application of these methods on learning the local structure in the conditional probability distributions with variable number of parameters that quantify these networks as opposed to attempting to learn the global structure at once was studied in Friedman & Goldszmidt (1998).

Both approaches are expensive to compute and do not focus on local modifications of networks. Our scoring scheme is much cheaper to compute when local modifications are desired and allows the assessment of the importance of individual edges on the global fitness of the network.

We examined this problem from the perspective of conditional entropy Simovici & Baraty (2008) by seeking a set of parents for a node that reduces the conditional entropy of that node in presence of its parents as much as possible. Our main interest in this paper is to evaluate the “importance” of a set of edges of a BNS in the presence of data by measuring the fitness loss of the BNS due to pruning the set of edges. The evaluation is obtained starting from the Kullback-Leibler divergence between two probability distributions.

Later, we examine the relationship of the fitness loss measure introduced in this article and the conditional entropy, in particular, with the measure introduced in Simovici & Baraty (2008). Finally, we combine the two measures to get a new formula and justify its use to simplify a BNS that represents expert’s prior knowledge of the domain without considering the data.

Let \mathcal{D} be a data set and let us denote by $\mathbf{Attr}(\mathcal{D})$ its set of attributes. For $K = \{A_{i_1}, \dots, A_{i_k}\} \subseteq \mathbf{Attr}(\mathcal{D})$ let $I_K = \{i_1, \dots, i_k\}$ be its index set.

If $\mathbf{A} = (A_1, A_2, \dots, A_n)$ is a permutation of $\mathbf{Attr}(\mathcal{D})$, let \mathbf{A}_{I_K} be the sequence of attributes of set K ordered according to \mathbf{A} . Denote by $\text{Dom}(\mathbf{A}_{I_K})$ the Cartesian product of the domains of the attributes in the sequence \mathbf{A}_{I_K} , that is,

$$\text{Dom}(\mathbf{A}_{I_K}) = \text{Dom}(A_{i_1}) \times \dots \times \text{Dom}(A_{i_k}).$$

For $\mathbf{a} = (a_1, \dots, a_k) \in \text{Dom}(\mathbf{A}_{I_K})$ we denote by $\mathbf{A}_{I_K} = \mathbf{a}$, the event

$$A_{i_1} = a_1, \dots, A_{i_k} = a_k.$$

A BNS for data set \mathcal{D} is a structure \mathcal{B}_s with set of nodes $\mathcal{V}_s = \mathbf{Attr}(\mathcal{D})$ and set of edges $\mathcal{E}_s \subseteq \mathbf{Attr}(\mathcal{D}) \times \mathbf{Attr}(\mathcal{D})$. The attributes of the data set are treated as random variables. The BNS represents probabilistic dependencies among these attributes.

We denote by $\text{BNS}(\mathcal{D})$ the set of all possible structures for \mathcal{D} and by $\text{BNS}_{\mathbf{A}}(\mathcal{D})$ the set of all structures of $\text{BNS}(\mathcal{D})$ which only contain edges (A, A') such that A precedes A' in the permutation \mathbf{A} . If $\mathcal{B}_s \in \text{BNS}_{\mathbf{A}}(\mathcal{D})$, then $\text{Par}_{\mathcal{B}_s}^{\mathbf{A}}(i)$ is the index set of the set of parents of $A_i \in \mathbf{Attr}(\mathcal{D})$ in \mathcal{B}_s according to \mathbf{A} . This notation is extended to sets of nodes as follows. If $V \subseteq \mathcal{V}_s$ and I_V is the corresponding index set of V , then $\text{Par}_{\mathcal{B}_s}^{\mathbf{A}}(I_V)$ is $\cup_{i \in I_V} \text{Par}_{\mathcal{B}_s}^{\mathbf{A}}(i)$. The \mathcal{B}_s subscript and \mathbf{A} superscript are omitted when it is clear from context.

The BNS in $\text{BNS}_{\mathbf{A}}(\mathcal{D})$ that contains the maximum number of edges is called the *complete BNS* for sequence \mathbf{A} , denoted by $\mathcal{B}_{cs}^{\mathbf{A}}$, and is depicted in Figure 1.

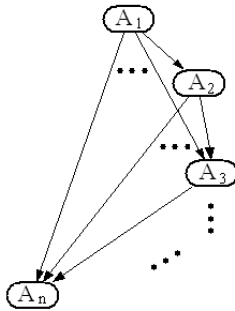


Figure 1: The complete BNS for ordering \mathbf{A} .

We make two basic assumptions:

1. The joint probability on the attributes of \mathcal{D} can accurately be represented by a BNS $\mathcal{B}_s^{\max} \in \text{BNS}_{\mathbf{A}}(\mathcal{D})$ and such a structure maximizes the posterior probability of the structure conditioned upon the data set.
2. An uniform prior probability distribution exists on all possible Bayesian network structures for \mathcal{D} .

Under these assumptions, $\mathcal{B}_{cs}^{\mathbf{A}}$ has maximum posterior probability in the presence of data.

For any $\mathcal{B}_s \in \text{BNS}(\mathcal{D})$ we have

$$P(\mathcal{B}_s | \mathcal{D}) \cdot P(\mathcal{D}) = P(\mathcal{D} | \mathcal{B}_s) \cdot P(\mathcal{B}_s)$$

by Bayes' Theorem. Since \mathcal{D} is fixed, and we assume $P(\mathcal{B}_s)$ is uniform, $P(\mathcal{B}_s | \mathcal{D})$ is proportional to $P(\mathcal{D} | \mathcal{B}_s)$. Thus, it suffices to show that

$$\frac{P(\mathcal{D} | \mathcal{B}_{cs}^{\mathbf{A}})}{P(\mathcal{D} | \mathcal{B}_s^{\max})} = 1.$$

Now, if we assume $\mathcal{D} = \{t_1, \dots, t_d\}$, by independence assumption of tuples of data set we have,

$$\frac{P(\mathcal{D} | \mathcal{B}_{cs}^{\mathbf{A}})}{P(\mathcal{D} | \mathcal{B}_s^{\max})} = \frac{\prod_{i=1}^d P(t_i | \mathcal{B}_{cs}^{\mathbf{A}})}{\prod_{i=1}^d P(t_i | \mathcal{B}_s^{\max})}.$$

and by Bayesian network split of joint probability we have,

$$\frac{\prod_{i=1}^d P(t_i | \mathcal{B}_{cs}^{\mathbf{A}})}{\prod_{i=1}^d P(t_i | \mathcal{B}_s^{\max})} = \frac{\prod_{i=1}^d \prod_{j=1}^n P(A_j = t_i[j] | \mathbf{A}_{\text{Par}_{\mathcal{B}_{cs}^{\mathbf{A}}}(j)}} = t_i[\text{Par}_{\mathcal{B}_{cs}^{\mathbf{A}}}(j)])}{\prod_{i=1}^d \prod_{j=1}^n P(A_j = t_i[j] | \mathbf{A}_{\text{Par}_{\mathcal{B}_s^{\max}}(j)}} = t_i[\text{Par}_{\mathcal{B}_s^{\max}}(j)])}$$

where if t is a tuple in \mathcal{D} and L is a set of attributes, then we denote the restriction of the tuple t to L be $t[L]$; we refer to $t[L]$ as the *projection* of t on L . Occasionally, we use $t[I_L]$ instead of $t[L]$, where I_L is the index set of L .

A Bayesian network structure for a data set incorporates a collection of conditional independence properties among attributes of that data set. This is captured by the *directed Markov property* of Bayesian networks Cowell (1998). This property stipulates that for any node X we have:

$$P(X | \text{nd}(X), \text{Par}(X)) = P(X | \text{Par}(X)), \quad (1)$$

which is denoted with $X \perp \text{nd}(X) \mid \text{Par}(X)$, where $\text{nd}(X)$ is the set of non-descendent nodes of X . Note that $\text{Par}_{\mathcal{B}_s^{\max}}(j) \subseteq \text{Par}_{\mathcal{B}_{cs}^{\mathbf{A}}}(j)$ for $1 \leq j \leq n$. Then, since we assumed \mathcal{B}_s^{\max} accurately represents the distribution over $\mathbf{Attr}(\mathcal{D})$ and by directed Markov property we have,

$$\begin{aligned} & P(A_j = t_i[j] | \mathbf{A}_{\text{Par}_{\mathcal{B}_{cs}^{\mathbf{A}}}(j)}} = t_i[\text{Par}_{\mathcal{B}_{cs}^{\mathbf{A}}}(j)]) \\ &= P(A_j = t_i[j] | \mathbf{A}_{\text{Par}_{\mathcal{B}_s^{\max}}(j)}} = t_i[\text{Par}_{\mathcal{B}_s^{\max}}(j)], \mathbf{A}_{\mathcal{C}} = t_i[\mathcal{C}]) \\ &= P(A_j = t_i[j] | \mathbf{A}_{\text{Par}_{\mathcal{B}_s^{\max}}(j)}} = t_i[\text{Par}_{\mathcal{B}_s^{\max}}(j)]) \\ & \quad \text{(by Markov property)} \end{aligned}$$

for all i and j where $\mathcal{C} = \text{Par}_{\mathcal{B}_{cs}^{\mathbf{A}}}(j) - \text{Par}_{\mathcal{B}_s^{\max}}(j)$. This justifies our proposition.

Yet, the complexity of a complete structure for a given sequence makes any computation prohibitively expensive. This demands the introduction of a measure which allows the simplification of the structure without incurring a significant loss of fitness. Such a measure may also be used to incrementally modify a BNS as new data becomes available.

2 Entropy and Partitions

A *partition* of a set S is non-empty collection of non-empty subsets of S , $\pi = \{B_i | i \in I\}$, such that $\bigcup_{i \in I} B_i = S$ and $B_i \cap B_j = \emptyset$ for all $i, j \in I$ where $i \neq j$. The set of partitions of a set S is denoted by $\text{PART}(S)$.

A partial order relation on $\text{PART}(S)$ is defined by $\pi \leq \sigma$ for $\pi, \sigma \in \text{PART}(S)$ where $\sigma = \{C_1, C_2, \dots, C_n\}$, if every block B_i of π is included in a block C_j of σ . The partially ordered set $(\text{PART}(S), \leq)$ is actually a bounded lattice. The infimum of two partitions π and $\pi' = \{B_j | j \in J\}$ on S , denoted with $\pi \wedge \pi'$, is the partition $\{B_i \cap B_j | i \in I, j \in J, B_i \cap B_j \neq \emptyset\}$ on S . The least element of this lattice is the partition $\alpha_S = \{\{s\} \mid s \in S\}$; the largest is the partition $\omega_S = \{S\}$.

The notion of entropy for partitions of finite sets was and axiomatized in Simovici & Jaroszewicz (2002). If S

is a finite set and $\pi = \{B_1, \dots, B_m\}$ is a partition of S , the entropy of π is the number

$$\mathcal{H}(\pi) = - \sum_{i=1}^m \frac{|B_i|}{|S|} \log_2 \frac{|B_i|}{|S|}.$$

Clearly, this is the Shannon entropy of a probability distribution (p_1, \dots, p_m) , where $p_i = \frac{|B_i|}{|S|}$ for $1 \leq i \leq m$. The main advantage of using partitions rather than probability distributions is the possibility of using the partial order defined on $\text{PART}(S)$. The following statement, proven in Simovici & Jaroszewicz (2002) is used in the sequel.

Theorem 2.1 *The entropy $\mathcal{H} : \text{PART}(S) \rightarrow \mathbb{R}_{\geq 0}$ is anti-monotonic; in other words, if $\pi \leq \pi'$, then $\mathcal{H}(\pi) \geq \mathcal{H}(\pi')$ for every $\pi, \pi' \in \text{PART}(S)$.*

The trace of a partition π on a subset T of S is the partition $\pi_T = \{T \cap B_i \mid i \in I \text{ and } T \cap B_i \neq \emptyset\}$ of T . Let $\pi, \sigma \in \text{PART}(S)$ be two partitions, where $\pi = \{B_1, \dots, B_m\}$ and $\sigma = \{C_1, \dots, C_n\}$. The entropy of π conditioned on σ is the number:

$$\mathcal{H}(\pi|\sigma) = \sum_{j=1}^n \frac{|C_j|}{|S|} \mathcal{H}(\pi_{C_j}).$$

It is immediate that $\mathcal{H}_\beta(\pi|\omega_S) = \mathcal{H}_\beta(\pi)$ and that $\mathcal{H}(\pi|\alpha_S) = 0$. Also, in Simovici & Jaroszewicz (2006) it is shown that $\mathcal{H}_\beta(\pi|\sigma) = \mathcal{H}_\beta(\pi \wedge \sigma) - \mathcal{H}_\beta(\sigma)$, a property that extends the similar property of Shannon entropy.

The next theorem proven in Simovici & Jaroszewicz (2002), Simovici (2007) states that conditional entropy is anti-monotonic with respect to its first argument and is monotonic with respect to its second argument.

Theorem 2.2 *Let $\pi, \sigma, \sigma' \in \text{PART}(S)$, where S is a finite set. If $\sigma \leq \sigma'$, then $\mathcal{H}(\sigma|\pi) \geq \mathcal{H}(\sigma'|\pi)$ and $\mathcal{H}(\pi|\sigma) \leq \mathcal{H}(\pi|\sigma')$.*

Finally, we mention the following corollary, also proven in Simovici & Jaroszewicz (2002).

Corollary 2.3 *Let S be a finite set. For every $\pi, \sigma \in \text{PART}(S)$ we have $\mathcal{H}(\pi|\sigma) \leq \mathcal{H}(\pi)$.*

Definition 2.4 The equivalence relation “ \sim^{A_I} ” defined by the sequence of attributes \mathbf{A}_I on \mathcal{D} , consists of those pairs $(t, t') \in \mathcal{D}^2$ such that $t[\mathbf{A}_I] = t'[\mathbf{A}_I]$.

The corresponding partition $\pi^{A_I} \in \text{PART}(\mathcal{D})$ is the partition generated by \mathbf{A}_I . \square

It is clear that if $I' \subseteq I$ then $\pi^{A_I} \leq \pi^{A_{I'}}$.

3 A Distribution Distortion Measure

Let us denote by $\mathbf{p}_{I_V}^{\text{Par}(I_V)}(\mathbf{a})$ the conditional probability distribution,

$$(P(\mathbf{A}_{I_V} = \mathbf{b}_1 | \mathbf{A}_{\text{Par}(I_V)} = \mathbf{a}), \dots, P(\mathbf{A}_{I_V} = \mathbf{b}_m | \mathbf{A}_{\text{Par}(I_V)} = \mathbf{a})) ,$$

where $\text{Dom}(\mathbf{A}_{I_V}) = \{\mathbf{b}_1, \dots, \mathbf{b}_m\}$ and $\mathbf{a} \in \text{Dom}(\mathbf{A}_{\text{Par}(I_V)})$.

To avoid unnecessary complications we assume $I_V \cap \text{Par}(I_V) = \emptyset$ in what follows, although the results that we have hold without this condition.

Let $E = \{(A_{s_1}, A_{d_1}), \dots, (A_{s_r}, A_{d_r})\}$ be a subset of the set \mathcal{E} of the edges of \mathcal{B}_s and let $S_E = \{s_1, \dots, s_r\}$ be the set of source nodes of edges of E and $D_E = \{d_1, \dots, d_r\}$ be the set of destination nodes for E . We

assume that $\text{Par}(D_E) \cap D_E = \emptyset$. Note also that $S_E \subseteq \text{Par}(D_E)$.

Clearly, if we remove the set of edges E from \mathcal{B}_s , the effect of this pruning on the joint probability distribution of the data set represented by \mathcal{B}_s will only be through conditional distributions attached to nodes of D_E . Thus, to assess the effect of pruning the edges of E from \mathcal{B}_s on the joint probability distribution of attributes of data set, consider the conditional probability distribution $\mathbf{p}_{D_E}^{(\text{Par}(D_E) - S_E)}(\mathbf{a}')$,

$$(P(\mathbf{A}_{D_E} = \mathbf{b}_1 | \mathbf{A}_{(\text{Par}(D_E) - S_E)} = \mathbf{a}'), \dots, P(\mathbf{A}_{D_E} = \mathbf{b}_m | \mathbf{A}_{(\text{Par}(D_E) - S_E)} = \mathbf{a}')) ,$$

where $\text{Dom}(\mathbf{A}_{D_E}) = \{\mathbf{b}_1, \dots, \mathbf{b}_m\}$ and $\mathbf{a}' \in \text{Dom}(\mathbf{A}_{(\text{Par}(D_E) - S_E)})$.

Note that $\mathbf{p}_{D_E}^{(\text{Par}(D_E) - S_E)}(\mathbf{a}')$ is the probability distribution of tuple \mathbf{A}_{D_E} conditioned on its set of parents after the removal of the set of edges E instantiated with \mathbf{a}' . To see how much the distribution is distorted as we prune the set of edges E from \mathcal{B}_s , we compare the probability distributions of \mathbf{A}_{D_E} conditioned on its set of parents in \mathcal{B}_s before and after removal of the set of edges for all possible instantiations of the sequence of parents.

For each instantiation $\mathbf{a}' \in \text{Dom}(\mathbf{A}_{(\text{Par}(D_E) - S_E)})$ of the sequence of parents of \mathbf{A}_{D_E} after pruning, there are several instantiations of the sequence of parents of \mathbf{A}_{D_E} before pruning, $\mathbf{a}_1, \dots, \mathbf{a}_z$, where $\mathbf{a}_i \in \text{Dom}(\mathbf{A}_{\text{Par}(D_E)})$ such that $\mathbf{a}_i[\text{Par}(D_E) - S_E] = \mathbf{a}'$ for $1 \leq i \leq z$. Thus, we need to compare the probability distributions

$$\mathbf{p}_{D_E}^{\text{Par}(D_E)}(\mathbf{a}) \text{ and } \mathbf{p}_{D_E}^{(\text{Par}(D_E) - S_E)}(\mathbf{a}'),$$

for $\mathbf{a} \in \text{Dom}(\mathbf{A}_{\text{Par}(D_E)})$ and $\mathbf{a}' \in \text{Dom}(\mathbf{A}_{(\text{Par}(D_E) - S_E)})$ such that $\mathbf{a}' = \mathbf{a}[\text{Par}(D_E) - S_E]$. Then, we can linearly combine the divergence between the pairs, weighted by the probability of occurrence of \mathbf{a} derived from data set \mathcal{D} .

To compare two finite probability distributions

$$\mathbf{p} = (p_1, p_2, \dots, p_n) \text{ and } \mathbf{q} = (q_1, q_2, \dots, q_n)$$

we use the *Kullback-Leibler* divergence measure given by

$$\text{KL}(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^n p_i \cdot \log_2 \frac{p_i}{q_i}.$$

KL has well-known properties:

1. $\text{KL}(\mathbf{p}, \mathbf{q}) \geq 0$ for all finite probability distributions \mathbf{p} and \mathbf{q} .
2. $\text{KL}(\mathbf{p}, \mathbf{q}) = 0$ if and only if $\mathbf{p} = \mathbf{q}$ element-wise.

However, $\text{KL}(\mathbf{p}, \mathbf{q})$ has no upper bound which makes comparisons for realizing the level of differences among a set of distributions difficult. We overcome this problem by dividing our linearly weighted measure of differences among the conditional probability distributions before and after edge pruning by the same weighted measure, but this time among the conditional probability distributions before the edge removal and the non-informative uniform probability distribution $\mathbf{u}_m \in [0, 1]^m$:

$$\mathbf{u}_m = \left(\frac{1}{m}, \frac{1}{m}, \dots, \frac{1}{m} \right),$$

where $m = |\text{Dom}(\mathbf{A}_{D_E})|$. If $\mathbf{p}_{D_E}^{\text{Par}(D_E)}(\mathbf{a}) = \mathbf{u}_m$ for all $\mathbf{a} \in \text{Dom}(\text{Par}(\mathbf{A}_{D_E}))$, then knowing the assignment of values to $\mathbf{A}_{\text{Par}(\mathbf{A}_{D_E})}$ is completely non-informative in predicting the value of \mathbf{A}_{D_E} . Also, assuming $|\mathcal{D}|$ is a multiple

of m , the m -block partition on \mathcal{D} which corresponds to the finite probability distribution \mathbf{u}_m , $\pi^{\mathbf{u}_m} = \{B_1, \dots, B_m\}$ where $|B_1| = \dots = |B_m| = \frac{|\mathcal{D}|}{m}$ is referred to as m -block uniform partition of \mathcal{D} and it has the maximum entropy, $\mathcal{H}(\pi^{\mathbf{u}_m}) = \log_2(m)$, over all possible partitions of \mathcal{D} with m blocks.

Definition 3.1 The distribution distortion caused by removing the set of edges E from the BNS \mathcal{B}_s denoted by $DD_{\mathcal{B}_s}(E)$ where $D_E \cap \text{Par}(D_E) = \emptyset$, is defined as

$$\frac{\sum_{\mathbf{a}} P(\mathbf{A}_{\text{Par}(D_E)} = \mathbf{a}) \cdot \text{KL}(\mathbf{p}_{D_E}^{\text{Par}(D_E)}(\mathbf{a}), \mathbf{p}_{D_E}^{Q_E}(\mathbf{a}[Q_E]))}{\sum_{\mathbf{a}} P(\mathbf{A}_{\text{Par}(D_E)} = \mathbf{a}) \cdot \text{KL}(\mathbf{p}_{D_E}^{\text{Par}(D_E)}(\mathbf{a}), \mathbf{u}_m)},$$

where D_E and S_E are defined as before and $Q_E = \text{Par}(D_E) - S_E$. Also the sums are over all $\mathbf{a} \in \text{Dom}(\mathbf{A}_{\text{Par}(D_E)})$. \square

Theorem 3.2 We have:

$$DD_{\mathcal{B}_s}(E) = \frac{\mathcal{H}(\pi^{\mathbf{A}_{D_E}} | \pi^{\mathbf{A}_{Q_E}}) - \mathcal{H}(\pi^{\mathbf{A}_{D_E}} | \pi^{\mathbf{A}_{\text{Par}(D_E)}})}{\mathcal{H}(\pi^{\mathbf{u}_m}) - \mathcal{H}(\pi^{\mathbf{A}_{D_E}} | \pi^{\mathbf{A}_{\text{Par}(D_E)}})},$$

where $m = |\text{Dom}(\mathbf{A}_{D_E})|$.

Proof. See Appendix A. \blacksquare

Corollary 3.3 We have $0 \leq DD_{\mathcal{B}_s}(E) \leq 1$.

Proof. Since $Q_E \subseteq \text{Par}(D_E)$, we have $\pi^{\mathbf{A}_{\text{Par}(D_E)}} \leq \pi^{\mathbf{A}_{Q_E}}$. By the monotonicity property of conditional entropy with respect to its second argument we have:

$$\mathcal{H}(\pi^{\mathbf{A}_{D_E}} | \pi^{\mathbf{A}_{\text{Par}(D_E)}}) \leq \mathcal{H}(\pi^{\mathbf{A}_{D_E}} | \pi^{\mathbf{A}_{Q_E}}).$$

Also,

$$\mathcal{H}(\pi^{\mathbf{A}_{D_E}} | \pi^{\mathbf{A}_{Q_E}}) \leq \mathcal{H}(\pi^{\mathbf{A}_{D_E}} | \{\mathcal{D}\}) = \mathcal{H}(\pi^{\mathbf{A}_{D_E}}).$$

But we know,

$$\mathcal{H}(\pi^{\mathbf{A}_{D_E}}) \leq \mathcal{H}(\pi^{\mathbf{u}_m}).$$

The result follows immediately. \blacksquare

Theorem 3.4 We have $DD_{\mathcal{B}_s}(E) = 0$ if and only if

$$P(\mathbf{A}_{D_E} = b_i | \mathbf{A}_{\text{Par}(D_E)} = \mathbf{a}) = P(\mathbf{A}_{D_E} = b_i | \mathbf{A}_{\text{Par}(D_E) - S_E} = \mathbf{a}[\text{Par}(D_E) - S_E])$$

for all i , $1 \leq i \leq m$, and $\mathbf{a} \in \text{Dom}(\mathbf{A}_{\text{Par}(T)})$.

Proof. Note that we implicitly assume that $P(\mathbf{A}_{\text{Par}(D_E)} = \mathbf{a}) \neq 0$ because, otherwise, $P(\mathbf{A}_{D_E} = b_i | \mathbf{A}_{\text{Par}(D_E)} = \mathbf{a})$ is undefined. The statement follows from the second property of the KL measure. \blacksquare

Theorem 3.5 Let E and E' be two sets of edges of BNS \mathcal{B}_s . Then, if $\text{Par}(D_E) \cap D_E = \emptyset$, $D_E = D_{E'}$ and $S_E \subseteq S_{E'}$, we have $DD_{\mathcal{B}_s}(E) \leq DD_{\mathcal{B}_s}(E')$.

Proof. Since $S_E \subseteq S_{E'}$, we have $Q_{E'} \subseteq Q_E$. Then, by the monotonicity property of conditional entropy with respect to second argument we have,

$$\mathcal{H}(\pi^{\mathbf{A}_{D_E}} | \pi^{\mathbf{A}_{Q_E}}) \leq \mathcal{H}(\pi^{\mathbf{A}_{D_E}} | \pi^{\mathbf{A}_{Q_{E'}}}).$$

The result follows immediately. \blacksquare

The global search for a set of edges to be pruned to simplify a BNS \mathcal{B}_s can be very expensive. Alternatively, we can examine local structures defined by a node

and its set of parent nodes along with the set of parent-child edges. That is, we can set D_E to be a set of single node, and consider different subsets of its set of parents as S_E . Then, we seek a subset such that the $DD_{\mathcal{B}_s}$ is close to zero. Since if we prune a set E of incoming edges at X such that $S_E \subseteq \text{Par}(X)$ and $DD_{\mathcal{B}_s}(E)$ is close to zero, then $\mathbf{p}_X^{\text{Par}(X)}(\mathbf{a}) \approx \mathbf{p}_X^{Q_E}(\mathbf{a}[Q_E])$ for all $\mathbf{a} \in \text{Dom}(\text{Par}(X))$ by Theorem 3.4. This, in turn implies $P(X | \mathbf{A}_{\text{Par}(X)}) \approx P(X | \mathbf{A}_{Q_E})$.

But, the directed Markov property implies

$$P(X | \mathbf{A}_{\text{Par}(X)}, \mathbf{A}_{\text{nd}(X)}) = P(X | \mathbf{A}_{\text{Par}(X)}).$$

Then, we have

$$\begin{aligned} & P(X | \mathbf{A}_{Q_E}) \\ & \approx P(X | \mathbf{A}_{\text{Par}(X)}, \mathbf{A}_{\text{nd}(X)}) \\ & = P(X | \mathbf{A}_{Q_E}, \mathbf{A}_{S_E}, \mathbf{A}_{\text{nd}(X)}) \end{aligned}$$

Thus, we have

$$X \perp (\mathbf{A}_{S_E}, \mathbf{A}_{\text{nd}(X)}) \mid \mathbf{A}_{Q_E}.$$

Finally, by symmetry and decomposition properties of conditional independence Pearl (1988) we have

$$X \perp \mathbf{A}_{\text{nd}(X)} \mid \mathbf{A}_{Q_E}.$$

Thus, the conditional independence property of a node of a structure is preserved if we prune a set of incoming edges of the node with distribution distortion measure close to zero.

In fact the parent-child fitness measure,

$$0 \leq \frac{\mathcal{H}(\pi^X | \pi^{\mathbf{A}_{\text{Par}(X)}})}{\mathcal{H}(\pi^X)} \leq 1 \quad (2)$$

introduced in Simovici & Baraty (2008) has some similarity with $DD_{\mathcal{B}_s}$.

We have shown that if this measure is close to zero, then $\text{Par}(X)$ is a suitable parent set for node X . Finding parent-child relationships among the attributes of \mathcal{D} such that the measure (2) is close to zero, increases the posterior probability of \mathcal{D} conditioned upon the inducing BNS \mathcal{B}_s . As stated before, an increase in this probability leads to an increase in the posterior probability of the structure conditioned on data (assuming a uniform prior on possible Bayesian network structures for a data set, as in Cooper & Herskovits (1993)). This happens because if we choose a set of parents, $\text{Par}(X)$ for a node X such that $\mathcal{H}(\pi^X | \pi^{\mathbf{A}_{\text{Par}(X)}})$ is close to zero, then for those $\mathbf{a} \in \text{Dom}(\mathbf{A}_{\text{Par}(X)})$ such that $P(\mathbf{A}_{\text{Par}(X)} = \mathbf{a})$ is non-trivial, the probability $P(X = x | \mathbf{A}_{\text{Par}(X)} = \mathbf{a})$ is close to 1 for some $x \in \text{Dom}(X)$ and close to 0 for all other $x' \in \text{Dom}(X) \setminus \{x\}$.

If $P(\mathbf{A}_{\text{Par}(X)} = \mathbf{a})$ is large and $P(X = x | \mathbf{A}_{\text{Par}(X)} = \mathbf{a}) \approx 1$, this implies that $P(\mathbf{A}_{\text{Par}(X)} = \mathbf{a}, X = x)$ is large. Thus, for a BNS \mathcal{B}_s that we obtain in this way we have

$$\begin{aligned} & P(\mathcal{D} \mid \mathcal{B}_s) \\ &= \prod_{i=1}^{|\mathcal{D}|} \prod_{X \in \text{Attr}(\mathcal{D})} P_{\mathcal{B}_s}(X = t_i[X] \mid \mathbf{A}_{\text{Par}(X)} = t_i[\text{Par}(X)]) \\ &= \prod_{X \in \text{Attr}(\mathcal{D})} \prod_{\substack{x \in \text{Dom}(X) \\ \mathbf{a} \in \text{Dom}(\mathbf{A}_{\text{Par}(X)})}} (P_{\mathcal{B}_s}(X = x \mid \mathbf{A}_{\text{Par}(X)} = \mathbf{a}))^{\mathcal{N}_{(X, \mathbf{A}_{\text{Par}(X)})}(x, \mathbf{a})}, \end{aligned}$$

where $\mathcal{N}_{(X, \mathbf{A}_{\text{Par}(X)})(x, \mathbf{a})}$ is the number of tuples t in \mathcal{D} with $t[X] = x$ and $t[\text{Par}(X)] = \mathbf{a}$. Having a BNS \mathcal{B}_s such that $P_{\mathcal{B}_s}(X = x \mid \mathbf{A}_{\text{Par}(X)} = \mathbf{a})$ is close to one for those pairs (x, \mathbf{a}) with large $\mathcal{N}_{(X, \mathbf{A}_{\text{Par}(X)})(x, \mathbf{a})}$ justifies the increase in posterior probability of \mathcal{D} .

4 Constructing a BNS for a Data Set

Recall that if E is a set of edges, then $Q_E = \text{Par}(D_E) - S_E$ is the set of nodes that remain parents of the nodes in D_E after the edges in E are removed.

Definition 4.1 Let $A_i \in \text{Attr}(\mathcal{D})$ for $1 \leq i \leq n$. Then, the total measure of fitness loss by pruning the set of converging edges E at node A_i in BNS \mathcal{B}_s , is the number,

$$\alpha \cdot \frac{\mathcal{H}(\pi^{A_i} | \pi^{A_{Q_E}}) - \mathcal{H}(\pi^{A_i} | \pi^{A_{\text{Par}(A_i)}})}{\log_2 |\text{Dom}(A_i)| - \mathcal{H}(\pi^{A_i} | \pi^{A_{\text{Par}(A_i)}})} + (1 - \alpha) \cdot \frac{\mathcal{H}(\pi^{A_i} | \pi^{A_{Q_E}}) - \mathcal{H}(\pi^{A_i} | \pi^{A_{\text{Par}(A_i)}})}{\mathcal{H}(\pi^{A_i})},$$

denoted by $\mathbf{FL}_\alpha(A_i, \mathcal{B}_s, E)$, where $0 \leq \alpha \leq 1$. \square

Clearly, this measure is always in the range $[0, 1]$. Note that the left component of the sum is the distribution distortion measure of pruning the set of incoming edges, E . The right component measures the decrease in reduction of entropy of node A_i in presence of its parents after pruning of the set E . Thus, if $\mathbf{FL}_\alpha(A_i, \mathcal{B}_s, E)$ is close to zero, then both components will be close to zero. The left component ensures that the conditional independence is preserved after the pruning of E , while the right component preserves the posterior probability of \mathcal{D} conditioned upon the structure. We can choose $\alpha = \frac{1}{2}$ if we have no preference over any of the two measures. Note that, if E is a set of converging edges at node A_i and $E' \subseteq E$ then, $\mathbf{FL}_\alpha(A_i, \mathcal{B}_s, E') \leq \mathbf{FL}_\alpha(A_i, \mathcal{B}_s, E)$. This enables us to use the heuristic method explained in Simovici & Baraty (2008) to find a structure that fits the given data from a complete network structure. This complete structure can be induced from a total order of attributes that represents the expert's knowledge of the domain.

5 Experimental Results

As our first experiment, we started with a Bayesian network Structure for Neapolitan Cancer data set with 5 attributes and 10000 rows, pruned different subsets of converging edges at a single node and computed the total measure of fitness loss for each pruning. Figure 2 visualizes these pruned structures and their relation with each other as a graph which we refer to as meta-graph to avoid confusion with the Bayesian graphs for the data set. Also, we refer to the edges and nodes of the meta-graph as meta-edges and meta-nodes, respectively. Each meta-node represents a BNS for Neapolitan data set and each edge, a pruning transformation. That is, the destination meta-node of a meta-edge is obtained by removing a subset of converging edges at a single node from the source meta-node of that meta-edge. Each meta-node is labeled with a letter from A to I.

Table 1 represents the scores for each meta-node in figure 2 based on two schemes MDL and C-H score.

Table 2 shows the total fitness loss of each meta-edge for parameter $\alpha = \frac{1}{2}$ and its two components, distribution distortion and entropy loss for each pruning of set of edges. The fitness loss measure is strongly correlated with both scoring schemes (C-H and MDL), which shows the usefulness of this measure for simplifying a Bayesian network structure. Also, our measure can be used to assess the importance of various edges. For example, the total fitness loss measure suggests that the edge that whose

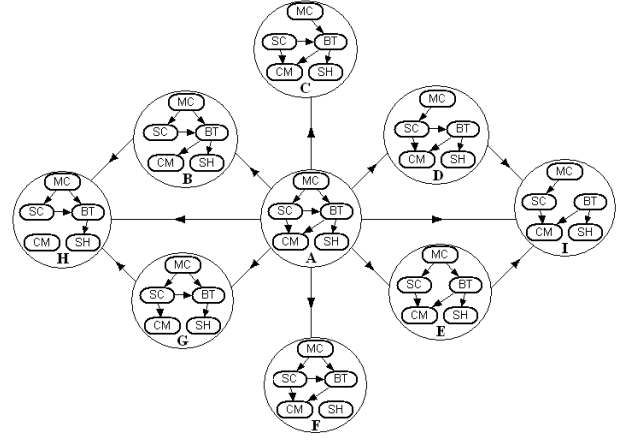


Figure 2: Neapolitan Cancer Bayesian structure was introduced in Cooper (1984). This structure has five nodes: Metastatic Cancer (MC), Serum Calcium (SC), Brain Tumor (BT), Coma (CM) and Severe Headaches (SH). Meta-node E corresponds to this structure.

Table 1: Neapolitan Cancer Scores

Structure	log(C-H Score)	MDL Score
A	-7505	24948
B	-7586	25214
C	-7938	26384
D	-7509	24958
E	-8120	26986
F	-7505	24947
G	-7697	25581
H	-7713	25632
I	-8341	27719

source is *Serum Calcium* and destination is *Brain Tumor* (SC to BT) is important since its removal causes a significant degradation of both the C-H and MDL scores. On the other hand, TFL assess the edge from Brain Tumor to Severe Headaches (SH) as not important. Again CH and MDL scores confirm this assessment.

Table 2: Neapolitan Cancer Pruning Measures with $\alpha = \frac{1}{2}$.

Edge	DD	Ent. Loss	TFL
AB	0.0526	0.092	0.0723
AC	0.684	0.2067	0.4454
AD	0.0068	0.0035	0.005
AE	0.5777	0.2976	0.4377
AF	0.023	0.001	0.012
AG	0.12155	0.2126	0.167
AH	0.1328	0.2322	0.1825
AI	0.786	0.405	0.5957
BH	0.0846	0.1402	0.1124
GH	0.0128	0.0196	0.0162
DI	0.7848	0.4016	0.593
EI	0.4938	0.1074	0.301

We also applied our approach on a Bayesian network structure for the ALARM data set, originally described in Beinlich et al. (1988) as a network for monitoring patients in intensive care. Table 3 contains scores for this structure which is labeled as A. Structures B to H are generated from A by pruning different subsets of parents for three nodes selected at random. Table 4 shows the exact parent pruning specification applied to obtain B to H from A. The child column represents the node we have chosen to prune its incoming edges. The original parent column shows the set of parents in the original structure, namely

A. New parent column represents the set of parents of the child after pruning. The other columns are the same as in previous example. Note that again, there is a close correlation between fitness loss measure and different scores. Since the structures are much larger than in previous experiment, pruning an edge or two has a milder effect on the magnitude of the scores of the global structure than in the Neapolitan case for about the same total fitness loss which is a local measure.

Table 3: Alarm Scores

Structure	log(C-H Score)	MDL Score
A	-159636	530806
B	-164287	546157
C	-162785	541189
D	-161372	536491
E	-161731	537644
F	-161684	537514
G	-159767	531136
H	-159638	530802

Table 4: ALARM pruning measures with parameter $\alpha = \frac{1}{2}$. The nodes of the ALARM network are traditionally numbered from 1 though 37 in the literature. The correspondence between the node numbers mentioned in the table and the real attributes are as follows (8, HREKG), (9, HRSat), (27, Catecholamine), (29, Heart Rate) and (30, Error Caution).

Struct.	Child	O. Par	N. Par	DD	Ent. Loss	TFL
B	8	30,29	none	0.4388	0.778	0.6083
C	8	30,29	30	0.297	0.5266	0.4118
D	8	30,29	29	0.1655	0.293	0.2294
E	9	30,29	none	0.2847	0.319	0.3018
F	9	30,29	30	0.2767	0.31	0.2933
G	9	30,29	29	0.0212	0.0237	0.0225
H	29	27	none	0.0006	0.001	0.0008

Finally, Table 5 shows the correlations between TFL and changes in logarithm of CH score and also between TFL and changes in MDL score for Neapolitan Cancer and ALARM structures as a result of edge removals explained in Tables 2 and 4. Interestingly, although TFL is a local measure, it has very close correlations with MDL and CH scores which are global measures.

Also note that while the correlations between TFL and MDL are positive, the correlations between TFL and CH score are negative, since as TFL increases, the probability of the structure conditioned upon the data set decreases and as a result the CH score decreases.

Table 5: Correlations

Data Set	TFL/log(CH)	TFL/MDL
Neapolitan	-0.97324	0.9733857
ALARM	-0.9983168	0.9980918

6 Conclusions and Future Work

We proposed a method for assessing the degree of influence of a set of edges of a Bayesian network structure on local conditional probability distributions. In particular, for the purpose of constructing a BNS from data, we concentrate on pruning a set of converging edges at a single node. This local pruning has a direct effect on the global fitness of the Bayesian network structure, measured by scoring schemes such as MDL or CH, which appear to be strongly correlated to the distribution distortion proposed by us. Thus, pruning is useful for adjusting a Bayesian network structure obtained from an expert's prior knowledge of the domain to a data set.

The distribution distortion could be used as measure of importance and interestingness of the edges of the Bayesian network structure and we intend to further pursue this issue. Another open technical problem is to explore whether by pruning a complete Bayesian network structure in the presence of a data set can lead to a network structure that best fits the data.

A Proof of Theorem 3.2

We substitute the Kullback-Leibler measure in the numerator,

$$\begin{aligned}
& \sum_{\mathbf{a} \in \text{Dom}(\mathbf{A}_{\text{Par}(D_E)})} P(\mathbf{A}_{\text{Par}(D_E)} = \mathbf{a}) \cdot \sum_{i=1}^m \left[P(\mathbf{A}_{D_E} = b_i | \mathbf{A}_{\text{Par}(D_E)} = \mathbf{a}) \right. \\
& \quad \left. \cdot \log_2 \frac{P(\mathbf{A}_{D_E} = b_i | \mathbf{A}_{\text{Par}(D_E)} = \mathbf{a})}{P(\mathbf{A}_{D_E} = b_i | \mathbf{A}_{Q_E} = \mathbf{a}[Q_E])} \right] \\
&= \sum_{\mathbf{a} \in \text{Dom}(\mathbf{A}_{\text{Par}(D_E)})} \sum_{i=1}^m \left[P(\mathbf{A}_{D_E} = b_i, \mathbf{A}_{\text{Par}(D_E)} = \mathbf{a}) \right. \\
& \quad \left. \cdot \log_2 P(\mathbf{A}_{D_E} = b_i | \mathbf{A}_{\text{Par}(D_E)} = \mathbf{a}) \right] \\
&- \sum_{\mathbf{a} \in \text{Dom}(\mathbf{A}_{\text{Par}(D_E)})} \sum_{i=1}^m \left[P(\mathbf{A}_{D_E} = b_i, \mathbf{A}_{\text{Par}(D_E)} = \mathbf{a}) \right. \\
& \quad \left. \cdot \log_2 P(\mathbf{A}_{D_E} = b_i | \mathbf{A}_{Q_E} = \mathbf{a}[Q_E]) \right] \\
&= -\mathcal{H}(\pi^{\mathbf{A}_{D_E}} | \pi^{\mathbf{A}_{\text{Par}(D_E)}}) \\
&- \sum_{\substack{\mathbf{a}' \in \text{Dom}(\mathbf{A}_{Q_E}) \\ \mathbf{a}'' \in \text{Dom}(\mathbf{A}_{S_E})}} \sum_{i=1}^m \left[P(\mathbf{A}_{D_E} = b_i, \mathbf{A}_{Q_E} = \mathbf{a}', \mathbf{A}_{S_E} = \mathbf{a}'') \right. \\
& \quad \left. \cdot \log_2 P(\mathbf{A}_{D_E} = b_i | \mathbf{A}_{Q_E} = \mathbf{a}') \right] \\
&= -\mathcal{H}(\pi^{\mathbf{A}_{D_E}} | \pi^{\mathbf{A}_{\text{Par}(D_E)}}) \\
&- \sum_{\mathbf{a}' \in \text{Dom}(\mathbf{A}_{Q_E})} \sum_{i=1}^m \left[\log_2 P(\mathbf{A}_{D_E} = b_i | \mathbf{A}_{Q_E} = \mathbf{a}') \right. \\
& \quad \left. \cdot \sum_{\mathbf{a}'' \in \text{Dom}(\mathbf{A}_{S_E})} P(\mathbf{A}_{D_E} = b_i, \mathbf{A}_{Q_E} = \mathbf{a}', \mathbf{A}_{S_E} = \mathbf{a}'') \right] \\
&= -\mathcal{H}(\pi^{\mathbf{A}_{D_E}} | \pi^{\mathbf{A}_{\text{Par}(D_E)}}) - \sum_{\substack{\mathbf{a}' \in \text{Dom}(\mathbf{A}_{Q_E}) \\ i \in [1..m]}} \left[\log_2 P(\mathbf{A}_{D_E} = b_i | \mathbf{A}_{Q_E} = \mathbf{a}') \right. \\
& \quad \left. \cdot P(\mathbf{A}_{D_E} = b_i, \mathbf{A}_{Q_E} = \mathbf{a}') \right] \\
&= \mathcal{H}(\pi^{\mathbf{A}_{D_E}} | \pi^{\mathbf{A}_{Q_E}}) - \mathcal{H}(\pi^{\mathbf{A}_{D_E}} | \pi^{\mathbf{A}_{\text{Par}(D_E)}}).
\end{aligned}$$

In the same way we can show,

$$\begin{aligned}
& \sum_{\mathbf{a} \in \text{Dom}(\mathbf{A}_{\text{Par}(D_E)})} P(\mathbf{A}_{\text{Par}(D_E)} = \mathbf{a}) \cdot \text{KL}(\mathbf{P}_{D_E}^{\text{Par}(D_E)}(\mathbf{a}), \mathbf{u}_m) \\
&= \sum_{\substack{\mathbf{a} \in \text{Dom}(\mathbf{A}_{\text{Par}(D_E)}) \\ i \in [1..m]}} \left[P(\mathbf{A}_{D_E} = b_i, \mathbf{A}_{\text{Par}(D_E)} = \mathbf{a}) \right. \\
& \quad \left. \cdot \log_2 P(\mathbf{A}_{D_E} = b_i | \mathbf{A}_{\text{Par}(D_E)} = \mathbf{a}) \right] + \log_2 m \\
&= \mathcal{H}(\pi^{\mathbf{u}_m}) - \mathcal{H}(\pi^{\mathbf{A}_{D_E}} | \pi^{\mathbf{A}_{\text{Par}(D_E)}}).
\end{aligned}$$

References

- Beinlich, I., Suermondt, J., Chavez, M. & Cooper, G. (1988), The alarm monitoring system: A case study with two probabilistic inference techniques for belief networks, in 'Second European Conference on Artificial Intelligence in Medicine', London.
- Cooper, G. F. (1984), NESTOR: A computer-based medical diagnosis aid that integrates casual and probabilistic knowledge, PhD thesis, Stanford University.
- Cooper, G. F. & Herskovits, E. (1993), A Bayesian method for the induction of probabilistic networks from data, Technical Report KSL-91-02, Stanford University, Knowledge System Laboratory.
- Cowell, R. (1998), *Learning in Graphical Models*, MIT Press, Cambridge, MA, USA, pp. 9–26.

- Friedman, N. & Goldszmidt, M. (1998), *Learning in Graphical Models*, MIT Press, Cambridge, MA, USA, pp. 421–459.
- Heckerman, D., Geiger, D. & Chickering, D. M. (1995), Learning Bayesian networks: The combination of knowledge and statistical data, in 'Machine Learning', pp. 197–243.
- Lam, W. & Bacchus, F. (1994), 'Learning Bayesian belief networks: An approach based on the MDL principle', *Computational Intelligence* **10**, 269–293.
- Pearl, J. (1988), *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers, Inc, San Mateo, CA.
- Rissanen, J. (1978), 'Modeling by shortest data description', *Automatica* **14**, 456–471.
- Simovici, D. A. (2007), 'On generalized entropies and entropy metrics', *Journal of Multiple-valued Logic and Soft Computing* **13**, 295–320.
- Simovici, D. A. & Baraty, S. (2008), Structure inference of Bayesian networks from data: A new approach based on generalized conditional entropy, in 'EGC', pp. 337–342.
- Simovici, D. A. & Jaroszewicz, S. (2002), 'An axiomatization of partition entropy', *Transactions on Information Theory* **48**, 2138–2142.
- Simovici, D. A. & Jaroszewicz, S. (2006), 'A new metric splitting criterion for decision trees', *International Journal of Parallel, Emergent and Distributed Systems* **21**, 239–256.
- Suzuki, J. (1999), 'Learning Bayesian belief networks based on the MDL principle: An efficient algorithm using the branch and bound technique', *IEICE Trans. Information and Systems* pp. 356–367.

Author Index

- Al-Naymat, Ghazi, 117
- Baig, Muzammil Mirza, 159
Bailey, James, 139
Bain, Michael, 151
Baraty, Saaïd, 193
Bhat, Savita S., 83
Buckley, Brian, 109
- Calvo, Rafael A., 175
Cao, Longbing, 63
Chan, Jeffrey, 139
Chawla, Sanjay, 117
Christen, Peter, iii
- Dai, Honghua, 183
Deshpande, Shailesh S., 83
Dew, Robert, 35, 75
- Gan, Min, 183
Gayler, Ross, 7
Guha, Sumanta, 11
- Homayounfard, Hooman, 129
Huang, BingQuan, 109
- Kan, Andrey, 139
Kaosar, Md. Golam, 17
Kechadi, Tahar, 109
Kelarev, Andrei, 25
Kennedy, Paul J., iii, 129
Koh, Yun Sing, 69
- Le Mercier, Nicolas, 167
Leckie, Christopher, 139
Lefait, Guillem, 109
Li, Jiuyong, 159, 167
Liu, Jixue, 159
Liu, Qing, 55
Liu, Yu-Hsn, 35, 75
Lock, Phillip, 167
- Ma, Liping, 25
- Murray, D. Wayne, 93
- Nankani, Ekta, 99
- Ofoghi, Bahadorreza, 25
Ong, Kok-Leong, iii
- Palshikar, Girish K., 83
Pears, Russel, 69
Pei, Jian, 5
Poon, Josiah, 43
- Ren, Yongli, 35, 75
Rong, Jia, 35
- Sato, Takeshi, 109
Shan, Yin, 93
Simoff, Simeon, 99
Simovici, Dan A., 193
Smith-Miles, Kate, 3
Stumptner, Markus, 167
Sutinen, Alison, 93
- Taheri, Javid, 117
- Vamplew, Peter, 25
Villalon, Jorge, 175
- Wang, Hua, 159
Webb, Dean, 25
Weng, Cheng G., 43
- Xu, Kai, 55
Xu, Meng, 151
Xu, Zhuojia, 17
- Yearwood, John, 25
Yi, Xun, 17
- Zhang, Ji, 55
Zhao, Yanchang, 63
Zheng, Zhigang, 63

Recent Volumes in the CRPIT Series

ISSN 1445-1336

Listed below are some of the latest volumes published in the ACS Series *Conferences in Research and Practice in Information Technology*. The full text of most papers (in either PDF or Postscript format) is available at the series website <http://crpit.com>.

Volume 79 - Conceptual Modelling 2008

Edited by Annika Hinze, *University of Waikato, New Zealand* and Markus Kirchberg, *Massey University, New Zealand*. January, 2008. 978-1-920682-60-6.

Contains the proceedings of the Fifth Asia-Pacific Conference on Conceptual Modelling (APCCM2008), Wollongong, NSW, Australia, January 2008.

Volume 80 - Health Data and Knowledge Management 2008

Edited by James R. Warren, Ping Yu, John Yearwood and Jon D. Patrick. January, 2008. 978-1-920682-61-3.

Contains the proceedings of the Australasian Workshop on Health Data and Knowledge Management (HDKM 2008), Wollongong, NSW, Australia, January 2008.

Volume 81 - Information Security 2008

Edited by Ljiljana Brankovic, *University of Newcastle* and Mirka Miller, *University of Ballarat*. January, 2008. 978-1-920682-62-0.

Contains the proceedings of the Australasian Information Security Conference (AISC 2008), Wollongong, NSW, Australia, January 2008.

Volume 82 - Grid Computing and e-Research

Edited by Wayne Kelly and Paul Roe *QUT*. January, 2008. 978-1-920682-63-7.

Contains the proceedings of the Australasian Workshop on Grid Computing and e-Research (AusGrid 2008), Wollongong, NSW, Australia, January 2008.

Volume 83 - Challenges in Conceptual Modelling

Edited by John Grundy, *University of Auckland, New Zealand*, Sven Hartmann, *Massey University, New Zealand*, Alberto H.F. Laender, *UFMG, Brazil*, Leszek Maciaszek, *Macquarie University, Australia* and John F. Roddick, *Flinders University, Australia*. December, 2007. 978-1-920682-64-4.

Contains the tutorials, posters, panels and industrial contributions to the 26th International Conference on Conceptual Modeling - ER 2007.

Volume 84 - Artificial Intelligence and Data Mining 2007

Edited by Kok-Loong Ong, *Deakin University, Australia*, Wenyuan Li, *University of Texas at Dallas, USA* and Junbin Gao, *Charles Sturt University, Australia*. December, 2007. 978-1-920682-65-1.

Contains the proceedings of the 2nd International Workshop on Integrating AI and Data Mining (AIDM 2007), Gold Coast, Australia. December 2007.

Volume 86 - Safety Critical Systems and Software 2007

Edited by Tony Cant, *Defence Science and Technology Organisation, Australia*. December, 2007. 978-1-920682-67-5.

Contains the proceedings of the 12th Australian Conference on Safety Critical Systems and Software, August 2006, Adelaide, Australia.

Volume 87 - Data Mining and Analytics 2008

Edited by Roddick, J.F., Li, J., Christen, P. and Kennedy, P.J., November 2008. 978-1-920682-68-2.

Contains the proceedings of the Seventh Australasian Data Mining Conference (AusDM 2008). Gold Coast, Australia.

Volume 90 - Advances in Ontologies.

Edited by Meyer, T. and Orgun, M.A., September 2008. 978-1-920682-71-2.

Contains the proceedings of the Knowledge Representation Ontology Workshop (KROW 2008), Sydney, Australia.

Volume 91 - Computer Science 2009.

Edited by Mans, B., January 2009. 978-1-920682-72-9.

Contains the proceedings of the Thirty-Second Australasian Computer Science Conference (ACSC 2009). Wellington, New Zealand.

Volume 92 - Database Technologies 2009.

Edited by Bouguettaya, A. and Lin, X., January 2009. 978-1-920682-73-6.

Contains the proceedings of the Twentieth Australasian Database Conference (ADC 2009). Wellington, New Zealand.

Volume 93 - User Interfaces 2009.

Edited by Weber, G. and Calder, P., January 2009. 978-1-920682-74-3.

Contains the proceedings of the Tenth Australasian User Interface Conference (AUIC 2009). Wellington, New Zealand.

Volume 94 - Theory of Computing 2009.

Edited by Downey, R. and Manyem, P., January 2009. 978-1-920682-75-0.

Contains the proceedings of the Fifteenth Computing: The Australasian Theory Symposium (CATS 2009). Wellington, New Zealand.

Volume 95 - Computing Education 2009.

Edited by Hamilton, M. and Clear, T., January 2009. 978-1-920682-76-7.

Contains the proceedings of the Eleventh Australasian Computing Education Conference (ACE 2009). Wellington, New Zealand.

Volume 96 - Conceptual Modelling 2009.

Edited by Kirchberg, M. and Link, S., January 2009. 978-1-920682-77-4.

Contains the proceedings of the Sixth Asia-Pacific Conference on Conceptual Modelling (APCCM 2009). Wellington, New Zealand.

Volume 97 - Health Informatics and Knowledge Management 2009.

Edited by Warren, J.R., January 2009. 978-1-920682-78-1.

Contains the proceedings of the Third Australasian Workshop on Health Informatics and Knowledge Management (HIKM 2009). Wellington, New Zealand.

Volume 98 - Information Security 2009.

Edited by Brankovic, L. and Susilo, W., January 2009. 978-1-920682-79-8.

Contains the proceedings of the Seventh Australasian Information Security Conference (AISC 2009). Wellington, New Zealand.

Volume 99 - Grid Computing and e-Research 2009.

Edited by Roe, P. and Kelly, W., January 2009. 978-1-920682-80-4.

Contains the proceedings of the Seventh Australasian Symposium on Grid Computing and e-Research (AusGrid 2009). Wellington, New Zealand.

Volume 100 - Safety Critical Systems and Software 2008.

Edited by Cant, T., December 2008. 978-1-920682-67-5.

Contains the proceedings of the Twelfth Australian Conference on Safety-Related Programmable Systems (SCS 2008). Canberra, Australia.